IN THE UNITED STATES PATENT
AND TRADEMARK OFFICE

32
pal
10-29

Merged Proceedings
Reissue Application No.:
09/512,592
United States Patent No.:
5,806,063
Issued: September 8, 1998
Applicant:
Dickens, Bruce M.
Reexamination Proceeding:
90/005,592
Filed: December 21, 1999
Reexamination Proceeding:
90/005,628
Filed: February 2, 2000
Reexamination Proceeding:
90/005,727
Filed: May 16, 2000
Response to Office Action

Group Art Unit: 2177

Examiner: Jean Homere

Box AF
Honorable Commissioner of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

Responsive to the Office Action dated April 16, 2003, the Applicant hereby submits the following:

DECLARATION OF EXPERT WITNESS

1. I Mark Winner am presently a software consultant to Boeing/McDonnell Douglas as a Senior Software Systems Engineer. In my current contract with Boeing the following business accounting systems have been my responsibility; Cost Charge Number System, Work Authority System, Merit Review System, Integrated Graphics Load, Labor Accounting and Reporting, Contract Status Information System, Manual Journals Vouchers, Rate Management System, Affordable Staffing, among others.

A large part of my responsibility has been for the migration of applications from old architecture to new Common Boeing Systems and new Web based applications. Part of this migration effort was to convert the HR systems to People Soft. I was involved in the selection and evaluation of the People Soft package, and the conversion and migration of applications to the Web. Labor accounting, cost charge number maintenance system, and reporting were redeveloped and deployed onto the Web. The department matrix for the company reorganization during the Rockwell/Boeing merge was also converted. I was also responsible for oversight of the company's Year 2000 Project, e.g., relating to accounting applications. This included the initial impact estimates of the project scope for accounting applications of the Expendable Launch Systems Division. Manpower estimates and requirements for project scope within corporate guidelines were provided. Also various organizations were worked with to determine if application processes could be achieved by other manual methods or migrated to other organizational Year 2000 compliant systems. Business critical and non-business critical systems with year 2000 risk had remediation plans written. Project statuses were managed, reported, and presented.

Systems and infrastructure were maintained, and users were supported with their ongoing needs. Support efforts were made to migrate and replace applications and hardware with newer technologies by migrating to Web based applications and client/server based applications were more cost effective. Technical support was provided by working with users in the system selection process or design and development of new Web based applications.

The above activities utilized the following languages: Cold Fusion, Java, Java Script, HTML, Dreamweaver, Visual Basic, Microsoft Basic, Access, Delphi, Crystal Reports, XML, Versata, ASP, Perl, ftp/tcpip/http, Versata and Rational Rose UML. Hardware and operating system platforms included NT, HP-UX/UNIX, IBM-AIX, HP3000 and IBM 3090's. Other Languages included Basic, Cobol, Fortran, ALC, TSO, VMS, CICS, Cognos, Protos, C, PL1, Pascal, SPF. Data Bases included Oracle 7.3, 8.0, 8I; Microsoft SQL Server; DB2 and Access DB.

In a prior employment at McDonnell Douglas I had Responsibility for maintenance of the Human Resources System including Compensation, Employee Biography, Employee Verification, Absence Tracking, Loan Verification, Employee Rate History, Ride Share, Merit Review, Seniority Training, Classified Materials Management System, and Medical Examination System. These systems were developed in Cobol and supported and enhanced with Supertool, Qedit, MPEX, Nbspool, Adager, MRJE, Allbase/SQL, Image/Query, NS3000 Communications and FTP. All above systems were network interfaced to be accessible to both local LAN systems and company corporate WAN's. Tools used included Novell NetWare 3.12, 4.00, 4.01. All systems were multi-server and fault tolerant. Communications were via dial-up and lease lines including TCP/IP.

A copy of my resume is attached to this Declaration.

2. I have studied United States Patent No. 5,806,063, issued to Dickens on September 8, 1998 on an application filed on October 3, 1996, entitled DATE FORMATTING AND SORTING FOR DATES SPANNING THE TURN OF THE CENTURY ("the Dickens patent"). I have also studied US. Patent No. 5,630,118, entitled SYSTEM AND METHOD FOR MODIFYING AND OPERATING A COMPUTER SYSTEM TO PERFORM DATE OPERATIONS ON DATE FIELDS SPANNING CENTURIES, filed on November 21, 1994 and issued on May 13, 1997 to Shaughnessy ("Shaughnessy"). I have also studied a translation of a published Japanese Application No. 05-027947, published on February 5, 1993, naming Masakazu Hazama as the inventor ("Hazama"). I have Studied the article of Ohms, B. *G. Ohms, Computer Processing of Dates Outside the Twentieth Century,* IBM Systems. Journal, Volume 25, Number 2, 1986, pages-244-51, ("Ohms"). In addition I am familiar with the Clipper 5 operating system as it existed on the date of the filing of the Dickens patent and have specifically reviewed the portions to which the Examiner in the above referenced Merged proceeding has made reference. I have also studied portions of the prosecution history of the Dickens patent before it was originally issued as mentioned in this Declaration. I have also studied the claims of the Dickens patent as originally issued and those added in the Reissue application in the

above referenced Merged Proceeding. I have in addition reviewed the content of Exhibit a filed with the dickens patent application.

4. I base my opinions expressed in this Declaration upon my knowledge of the art as a person of at least ordinary skill in the art at the time of the filing of the Dickens patent and on the above referenced materials which I have reviewed.

5. I understand the Dickens patent in its Specification, with or without the Exhibit A referred to in the body of the Specification to have disclosed to one of ordinary skill in the art to which it pertains at the time of its filing the following:

6. The Dickens patent notes, initially, that "[d]ates are stored as symbolic representations in computer data bases in varying formats." (Col. 1, lines 10-11) Examples of such formats are given as a "numerical representation MM/DD/YY, where MM is a two-digit month designator, DD is a two-digit day designator, and YY is a two-digit year designator (the last two digits of the year). ... A date may also be represented in an alphanumeric for MMM/DD/YY, where MMM is an alphanumeric month designator (e.g., DEC for December ... ." (Col. 1, lines 11-20)

7. Also notes the Dickens patent "[s]ets of dates spanning the turn of the century and associated with past, current, and future activities are now stored in many databases. When stored in the conventional formats discussed above, those dates will not readily be used and numerically sorted in chronological order." (Col. 1, lines 31-35)

8. Further, the Dickens patent notes that "Using the numerical form above, Dec. 15, 2000 is represented as 12/15/00. If a numerical sort is performed on 12/15/93 and 12/15/00, the later date 12/15/00 sorts as the first-occurring date, an incorrect result." (Col. 1, lines 28-30)

9. The Dickens patent also notes "[t]hey [the symbolic representations of dates suffering from the problem of the system being unable to distinguish dates 'spanning the turn of

the century'] may be manually converted to a more useable form in the sense that programs may be written to perform conversions, manipulations, and sorting. However, these programs typically require additional data fields for storage, which may be objectionable in some circumstances." (Col. 1, lines 35-40) The Dickens patent also notes "[t]he database includes information in the form of symbolic representations of dates and associated information such as events occurring on the respective dates." (Col. 2, lines 48-50)

10. As stated in the Dickens patent:

> The present invention provides an approach to the representation and utilization of dates stored symbolically [as defined above] in databases. Existing symbolic date representations [as defined above] are converted to a more useful form of symbolic date representations *without the addition of new data fields*, and in a manner that is performed automatically by the computer and requires no user input. (Col. 1, lines 49-55, Emphasis added)

11. The Dickens patent goes on to explain:

> a method of processing dates stored in a database [symbolically as described above] comprises the steps of providing a database with the dates stored therein according to a [symbolic] format [as discussed above, in which] Y_1Y_2 is the numerical year designator ... . A century designator C_1C_2 is *determined for each date in the database*, C_1C_2 ... . *Each date in the database is formatted* with the values C_1C_2 Y_1Y_2 ... . (Col. 1, line 57 – Col. 2, line 3, Emphasis added)

12. The Dickens patent also notes:

> The computer database 26 is provided, numeral 30, having symbolic representations of dates stored therein.
>
> ...
>
> A ten decade window is selected, numeral 32. That is, it is necessary that all dates in the database will be within some period of 10 decades, or 100 years.
>
> ...

The symbolic representations of the dates in the database are reformatted with the values C_1C_2 Y_1Y_2 ... . In one case that produces particularly advantageous results for many operations, such a chronological date sorting, the date is represented in the form C_1C_2 Y_1Y_2 ... .

Once the symbolic representations of the dates are reformatted according to the procedures set forth above, the date information may be sorted, numeral 38 or otherwise manipulated, numeral 40, together with the entries associated with the dates. Such manipulations may include handling of data associated with the dates, storing the dates back in the dates and information back in the data base, or other processes. (Col. 2, line 60 – Col. 3, line 55)

13. The Dickens patent further notes that:

the present invention thus provides an efficient approach to converting and utilizing symbolic date representations in databases [without requiring additional or modified data fields for storage in the existing database] which allows automatic processing of dates ranging from before to after the year 2000. The large number of dates represented in some databases may thereby be readily processed and utilized. (Col. 2, lines 22-27)

14. Claims 1 and 11 as originally filed recited:

"reformatting the symbolic representation of the date [in the database] with the values C_1C_2, Y_1Y_2, M_1M_2, D_1D_2," and "reformatting each date [in the database] in the form C_1C_2, Y_1Y_2, M_1M_2, D_1D_2" respectively.

15. The Examiner rejected these claims on the basis of lack of enablement, since:

[t]he 'conversion of existing symbolic date representations ... without the addition of new data fields', as indicated at page 2 lines 7-10, is critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. ... The problem set forth in the last four lines of page 1 and promised in the first paragraph of page 2, as well as in the lines quoted above, indicate that

the invention solves the Y2K problem without introducing additional digits. The claims, the abstract and the description of the invention in the SUMMARY clearly involve century digits C_1C_2, which increase the number of date digits from 6 to 8, thus using 4 digits to indicate the year. *One of ordinary skill in the art would not know how to resolve this discrepancy.* (Office Action of November 17, 1996, at page 2, Emphasis added)

16. The applicant responded in an Amendment of March 17, 1997. The applicant's counsel noted:

> As stated in the application, many existing databases contain date representations that are defined only by the decade and year designations (i.e., Y_1Y_2). Because these databases do not provide date designations for the century associated with each date, it will be impossible to discern the order of dates in a database after the turn of the century.
>
> To properly and efficiently address this problem, a method for converting dates in databases was needed. This method should accept dates from data storage, discern the proper century designation for each date, and reformat the dates with the century designation.
>
> The claimed invention provides a method of processing symbolic representations of dates stored in a database. ... Finally, the method of the claimed invention includes the step of reformatting the symbolic representations of the dates into corresponding values of C_1C_2, Y_1Y_2, M_1M_2, D_1D_2. These values can then be used to manipulate the dates, such as by sorting the dates in chronological order.
>
> In a typical operation of the claimed invention, the method performs date conversions on a database that includes dates from both the twentieth and twenty-first century. ...
>
> ... The method of the present invention further includes the step of reformatting the symbolic representations of the dates into values C_1C_2, Y_1Y_2, M_1M_2, D_1D_2. ... *These dates can then be used for several operations such as date sorting.*

[redacted] [redacted]

Advantageously, the method of the claimed invention can be implemented as an initial step in any database manipulation program. For instance, the method of the claimed invention may be embodied in computer software code that *preprocesses a database prior to beginning the remainder of the data manipulation program.* In this embodiment, the method initially converts the data from the varying formats, *determines the century designation for each date, and reformats the dates such that the dates may be used by the database manipulation program for such operations as sorting and printing*[1] the dates. In this embodiment of the present invention, *the dates are temporarily converted and reformatted for use by the manipulation program.* However the method of the present invention need not store the converted date in data storage. Instead, the *original dates in the data storage remain undisturbed.* This aspect of the present invention thus allows conversion of dates to compensate for century designations *without requiring the addition of data fields to permanently store the century designations.* (Emphasis added)

17. In addressing the specific rejection of the original Examiner applicant's counsel also notes:

> [T]he Office Action objects to the disclosure for implying that the current invention does not require additional data fields for storage to solve the year 2000 problem [and] rejected all of the claims ... as based on a disclosure that is not enabling. In particular the Office Action states that the conversion of the existing symbolic date representations without the addition of new data fields is critical or essential to the invention but not included in the claims.
>
> As described below, the method of the claimed invention does not require that the converted data that includes the century designations be stored in data storage. Likewise ... the amended set of claims does not require storage of the converted data and therefore imposes no requirement for new data fields. ...

[1] It is not apparent from this discussion of the disclosure of the Dickens patent whether applicant's counsel was relying also on the content of Exhibit A, but only Exhibit A in the disclosure as originally filed refers to performing a printing program after a sorting program. The Examiner did not object to this reference to performing a sorting program or a printing program.

[redacted]

As stated in the background of the invention, conventional date formatting systems typically require additional data fields for storage to accommodate the century designations. These additional data fields are necessary because conventional systems disclose a permanent reformatting of stored data. The claimed invention, on the other hand, does not require that the reformatted data be permanently stored. Instead, the method of [the] claimed invention encompasses embodiments in which the date information is initially reformatted and converted to have century designations, but does not require that the reformatted dates be stored. As stated previously, the method of one embodiment of the claimed invention reads the dates from the database and temporarily reformats the dates with century designations. Data manipulation programs are then performed on these reformatted dates, such as sorting the dates. However, once the data manipulations are complete, the reformatted dates need not be stored in data storage.

18. Disclosed in Shaughnessy is a "system and method for modifying and operating a computer system to perform date operations on date fields having a two digit representation for the year without erroneously mistaking the years 2000 *et seq.* for the years 1900 *et seq.*" (Col. 1, lines 11-14)

19. As a solution to this problem, Shaughnessy proposes:

> [i]n accordance with the present invention, the current date operation routines nested in the body of the application program would be replaced with a call to one of a plurality of subroutines stored externally from the existing application program, *as opposed to the date operation routine being reprogrammed to perform the date operation in a new format.* The subroutines will be able to *accommodate the date format currently employed* by the application program, thus making it *unnecessary to convert all of the data fields in files containing data used in the application program over to the new data format.* (Col 4, lines 27-38, emphasis added).

20. As an example Shaughnessy describes a program that would "perform[] a date comparison to determine when loan payments are overdue ... ." (Col.4, lines 39-40)

21. According to the Shaughnessy method, the:

> program statements which performed the above functions would be modified to include program statements which did the following:
>
> 1. *Call the subroutine* which performs the date comparison *passing today's date, the date the next payment is due, and a three byte parameter*, the first byte of which identifies the format of today's date, the second byte of which identifies the format of the date next payment is due, and the third byte of which is left available for a *return code indicative of the result of the comparison*;
>
> 2. *If the result received from the subroutine indicates that the date next payment is due is greater than today's date, indicate that the account is okay.* (Col 4, lines 48-62, emphasis added)

22. In order to do this, Shaughnessy suggests that "for the subroutines to be able to accommodate different date formats, certain information, namely the current date, end of 100 year cycle, and two possible century values, must be determined and made available to the subroutines." (Col. 4, line 66 - Col. 5, line 3) In addition, Shaughnessy teaches that "each subroutine that performs a date operation will include a call to another subroutine which can determine this information." (Col. 5, lines 3-5)

23. Further according to Shaughnessy "[t]he above-mentioned information will be used in the subroutine(s) to assign a century value to the two digit representation of the year of the dates to be operated on such that the subroutine can accurately perform its intended function." (Col. 5, lines 21-25) According to the Shaughnessy method this is performed using a form of windowing in which:

> [t]he current date is determined ... in a format which utilizes a four digit representation for the year. Initially, the current date is set to the operating system date in the format 00YYDDD ... by way of example ... 0094263 ... . The current date is then compared to the date the system was installed with the [date

conversion] modifications (modified system install date) ... which, for the sake of example, is 1994032 ... . If the YYDDD portion of the current date, 94263 is greater than or equal to the corresponding portion of this modified system install date, 94032 ... then the century of the current date is set to the century of the modified system install date ... ." ...

If the current date appeared less than the modified system install date ... in the 00YYDDD format ... then the current date century would be set in the format CCYYDDD to the century value for the modified system install date plus one ... . (Col 5, lines 31-57)

24. The Shaughnessy method then determines "the end of the 100 year cycle" according to "several parameters [which] may be specified." These "may include the number of years of future dating required (default is 10), ... and whether the end of the 100 year cycle is to be updated daily ... ." (Col. 6, lines 4-13) "If the cycle is to be updated daily, then the starting date is set to the current date ..., as determined above. ... Next, the end of the 100 year cycle is determined by adding the number of years future dating required to the starting date ... ." (Col 6, lines 17-22).

25. Further, explains Shaughnessy:

The application program currently operating in a particular computer system may have a comparison of two date fields as part of its operation. If so, the source code which performs the comparison can be replaced with a call to DS2000R1, the name given to an exemplary comparison subroutine useful in practicing the present invention ... .

...

As illustrated in FIG. 8, the call DS2000R1 ... is inserted into the application program, and includes parameters P1, P2, and P3. P1 and P2 are the date fields which are to be compared. For example, P1 could be "DATE-NEXT-PAYMENT-DUE" and P2 might be "TODAY" as referenced in the above sample of modified source code. P3 is a three byte field in which the first two bytes define the type of date field P1 and P2, respectively. The third byte is a return

code which will be set to a value indicative of the result of the comparison. (Col. 8, line 35 - Col. 9, line 53)

26. In summary, the teaching of Shaughnessy is to "[c]all the subroutine which performs the date [operation] passing [two dates] and the three byte parameter [including] a return code indicative of the result of the [operation]." Windowing occurs only in the called subroutine and in a manner other than that of the invention claimed in the Dickens patent as originally issued and/or as added in the Reissue application. Moreover, this "on call" or "on the fly" windowing of at most two date data entries at a time is not the subject matter of such invention.

27. There is, therefore, no teaching or suggestion in Shaughnessy of:

> selecting a 10-decade window with a Y_A Y_B value for the first decade of the window, Y_A Y_B being no later than the earliest Y_1 Y_2 year designator in the database; ... .

The Shaughnessy method selects a 10-decade window utilizing the "date the system was installed."

28. There is also, therefore, no teaching or suggestion in Shaughnessy of:

> determining a century designator C_1 C_2 for each symbolic representation of a date in the database, C_1 C_2 having ... ; ... .

The teaching of Shaughnessy is to determine a century designator for at most two date data representations being processed in a called subroutine at any given time.

29. There is also, therefore, no teaching or suggestion in Shaughnessy of:

> reformatting the symbolic representation of the date with the values C_1 C_2, Y_1 Y_2, M_1 M_2 , and D_1 D_2 to facilitate further processing of the dates.

The teaching of Shaughnessy is to reformat at most two dates at a time in the called subroutine and the return to the program from the called subroutine of an indicator of the result of the processing of the two reformatted date data entries. Shaughnessy does not

teach facilitating "further processing of the dates" by "reformatting the symbolic representation of the date" "for each symbolic representation of a date in the database."

30. There is accordingly also no teaching or suggestion in Shaughnessy of:

sorting the symbolic representations of dates; (claim 4).

The method of Shaughnessy does not teach sorting all of the "symbolic representations of dates." It teaches only the comparison of one date to a fixed date or two dates to each other in the called subroutine and returning to the program an indication of the result of the comparison.

31. There is also no teaching or suggestion in Shaughnessy of:

reformatting each symbolic representation of a date into the format C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2 (claim 5), nor sorting the symbolic representations of dates using a numerical-order sort (claim 6); nor storing the symbolic representation of dates and their associated information back into the database (claim 9), nor manipulating information in the database having the reformatted date information therein (claim 10).

32. In addition, there is no teaching or suggestion in Shaughnessy of:

converting pre-existing date information[within a database] having a different format into the format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator and Y_1 Y_2 is the numerical year designator (claim 7).

33. In addition, there is no teaching or suggestion in Shaughnessy of:

selecting Y_A Y_B such that Y_B is 0 (zero) (claim 8).

34. Japanese Published Patent Application, HEI 5-27947, entitled METHOD OF GUARANTEEING YEAR ORDER, with inventor Masakazu Hazama, published on February 5, 1993 ("Hazama") discloses a system "to guarantee the year order, even for

years after 2000 AD, with the current file format, even when the year is managed by the last two digits of the date in digital files." (Hazama, at 2)

35. The system of Hazama, like Shaughnessy, discussed above, modifies a date in a record using a "correspondence utility module (10)." (Hazama, at 2) For that single record "the position in the record where the last two digits AD had been previously stored are specified" to the module 10 from "external parameter 9." Hazama further notes that "the processing section will replace the code of the 10's place in the last two digits of the date with a code that maintains the year order." (Hazama, at 4)

36. A form of windowing is applied. Hazama notes that "the following processing is performed by the module (10)." (Hazama, at 5) This is then followed by "Work area output processing (5): data that have undergone replacement processing (4) are output to the work area (8)." (Hazama, at 6)

37. Referring to Figure 1 in Hazama, it is more clearly demonstrated that Hazama is not more applicable to the patentability of the claimed invention than Shaughnessy, as discussed above. The data from a record is moved from the processor "work area" 8 to the "Year 2000 date correspondence utility module" 10. A modified date, after some form of windowing, for that single record is returned to the work area 8 for processing.

38. Hazama, like Shaughnessy, therefore, does not disclose or suggest the claimed invention.

39. There is also, therefore, no teaching or suggestion in Hazama nor in the combination of Shaughnessy and Hazama of:

> determining a century designator C_1 C_2 for each symbolic representation of a date in the database, C_1 C_2 having ... ; ... .

The teaching of Hazama, or Shaughnessy in view of Hazama, is to determine a century designator for at most two date data representations being processed in a called subroutine/module at any given time.

40. There is also, therefore, no teaching or suggestion in Hazama, or Shaughnessy in view of Hazama, of:

> reformatting the symbolic representation of the date with the values C_1 C_2, Y_1 Y_2, M_1 M_2 , and D_1 D_2 to facilitate further processing of the dates.

The teaching of Shaughnessy, or Shaughnessy in view of Hazama, is to reformat at most two dates at a time in the called subroutine/module and the return to the program from the called subroutine/module of an indicator of the result of the processing of the two reformatted date data entries or a single modified date from a single record. Neither Shaughnessy nor Shaughnessy in view of Hazama teaches facilitating "further processing of the dates" by "reformatting the symbolic representation of the date" "for each symbolic representation of a date in the database."

41. Accordingly, there is also no teaching or suggestion in the combination of Shaughnessy and Hazama as applied by the Examiner of:

> sorting the symbolic representations of dates; (claim 4).

42. There is also accordingly no teaching or suggestion of:

> reformatting each symbolic representation of a date into the format C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2 (claim 5), nor sorting the symbolic representations of dates using a numerical-order sort (claim 6); nor storing the symbolic representation of dates and their associated information back into the database (claim 9), nor manipulating information in the database having the reformatted date information therein (claim 10).

43. In addition, there is accordingly no teaching or suggestion of:

> converting pre-existing date information [within a database] having a different format into the format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator and Y_1 Y_2 is the numerical year designator (claim 7).

44. In addition, there is accordingly no teaching or suggestion of:

selecting Y_A Y_B such that Y_B is 0 (zero) (claim 8).

45. Ohms teaches a "[m]ethod[] of using existing date formats across century boundaries ... . The use of a format termed the Lilian date format ... is introduced." (Ohms, at 244, Abstract) Ohms teaches that "[t]he two positions traditionally used in both Julian and Gregorian date formats implicitly represent a year within a century. However, this system is inadequate for representing dates in more than one century." (Id. at 245) As a solution Ohms proposes a "Lilian date format [to] avoid[] the ambiguity by using seven positions for the number of the days from the beginning of the Gregorian calendar, October 15, 1582." (Id. at 245) "The value is incremented by one for each subsequent day." (Id. at 246) Ohms explains that "the Lilian date format is presented here as the basis for making date conversions ... . This format handles processing across century years and other aspects of date conversion not currently adaptable to computer programming." (Id. at 244-45)

46. In this context, of database conversion to Lilian format from more traditional Gregorian or Julian formats, Ohms describes under the heading "Accommodating end users" the fact that they "usually enter two digits for the year in a date and understand the ambiguity that this represents." (Id. at 248) Ohms goes on to say that:

> to avoid adverse user reaction, [by requiring the entry of date data in other than two digits] programs must continue to function with only two digits for year. The inference of the year 1997 from 97 and 2003 from 03 must continue. For the exceptional case where the correct meaning could be 1897 and 1903, entry of all four digits may be required. (Id. at 248)

47. It is in this context also that Ohm notes:

> it may be necessary to provide a conversion function that receives a definition of the implied century as a parameter. An excellent way to do this unambiguously is to specify a year as the desired starting point of a 100-year range. For example, if the starting year for the range is specified as 1925, dates with year digits between

[redacted] [redacted]

25 and 99 would be between 1925 and 1999, and dates with year digits of 00 through 24 would lie between 2000 and 2024. (*Id.* at 248)

48. Ohms, or the combination of Ohms with either Hazama or Hazama along with Booth, therefore, simply teaches storing dates in a database in Lilian format which "handles processing across century years" and "[a]ccommodating end users" who "enter two digits for the year" by "providing a conversion function" using the known technique of windowing for data entry only.

49. Clipper 5A, described in Booth, in a fashion very similar to Ohms' use of the Lilian date format, as discussed below, operates with date data stored in databases representing dates in the form of a number, often referred to as integer date formatting. Each unique number represents a date or a time and date down to a specific time increment, e.g., a mili-second, starting with a certain date or date/time and counting to the limits of the number of binary places available, e.g., 32, incrementing the chosen time increments.[2] As noted, each integer could represent a day or day/time incremented by the value of the integer from a start date or day/time. That is to say, the total time period that can be represented depends upon the starting date or date/time, the number of unique combinations of, e.g., 32 bits, and the increment counted, e.g., days, seconds, mili-seconds, etc. [3]

---

[2] This is the same as the Lilian format disclosed in Ohms, however, only the date is stored in Lilian format, according to the teaching of Ohms, and the Lilian date format, strictly speaking, starts with the beginning of the Gregorian calendar. Ohms, however, does disclose a modified Lilian format starting at some other arbitrarily selected starting date. According to the discussion in Booth, at 949-951, Clipper may count integers for year/date separately from hours, minutes and seconds on a given date, but Booth discloses at 951 using a single number, an integer for the year/date, and a decimal component for the elapsed seconds from midnight on the particular year/date. This simply makes Clipper even more identical to Ohms' disclosure of a modified Lilian integer date with the integer representing some day incrementally counted from some starting day, spanning over several centuries. In this format, as noted, each incremental number represents a day or a moment in time, *each such day or moment being, by definition, a part of a fully defined and recoverable four digit year date datum.*

[3] Booth, at 939 ("Dates are stored internally in such a way that math operations can be performed on dates to derive other dates. Adding an integer to a date will result in a future date. Subtracting two dates will result in a number of days between the two.") Booth, at 99 ("The date type is used to represent calendar dates. Clipper stores dates internally in such a way that a variety of operations can be performed on them. You can determine the number of days between two dates by subtracting them, and you can determine a future date by adding an integer value to a date value. The result will be a date value, some number of days in the future.")

[redacted]

50.It is also true of dates stored in integer format, as is also the case with the dates stored in Lilian format, that there is no Y2K ambiguity problem in regard to dates stored in a database in these formats. Booth, therefore, like Ohms, does not even suffer from the problem that the claimed invention is meant to address. Knowing the starting point (the starting date in Lilian, or the starting date or date/time in integer), the granularity (i.e., one day in Lilian and, e.g., one day or one second or one mili-second in integer), and the incremental difference between the starting point and the integer date number stored in either the Lilian or integer format, the stored date is known, including the year to four digits. Therefore, included within what is so stored in the database is the information needed to determine a century designator. No possible Y2K ambiguity problem can exist or does exist when storing dates in a database in these formats. The claimed invention does not relate to databases with dates stored in these formats. The claimed invention involves databases with dates stored in them where there is an ambiguity because only two digits of date data are present in the stored information, from which to determine the full date, including a century designator. Every date stored in a database in Lilian or integer format, by definition, already has a century designator; can not possibly be ambiguous due to reaching the end of a century[4]; and never needs to have a century designator determined, whether by the method of the claimed invention or otherwise.

51. For this reason alone, Booth, like Ohms, has nothing to do with the invention as claimed in the Dickens patent other than providing another disclosure of a utilization of windowing with a ten decade window for a purpose unrelated to that for which such windowing Is used in the inventions as claimed in the claims of the Dickens patent as originally issued and as added in the Reissue applications. In fact, it teaches away from the claimed invention.

52. Booth describes a number of functions that the programming language utilizes to read or write dates into the database, display dates on a screen, find the difference between

[4] Even at the end of the time span that can be represented, e.g., as noted below Clipper dates run until December 31, 2999, there is no "ambiguity." The system simply cannot express a date beyond the given span (without changing the parameters, e.g., the length of the integer number or the granularity).

two dates or a date/time equal to a given date/time plus some incremental time period, to find the day of the week of a given date/time, and like functions. Like Ohms and Shaughnessy, some of these functions employ windowing in some fashion or another. Like Shaughnessy, when they do employ windowing these programming routines "[c]all the subroutine which performs the date [operation] passing [information] indicative of the result of the [operation]." Windowing occurs in the called subroutine in a manner other than that of the claimed invention. Such "on call" or "on the fly" windowing of at most two date data entries at a time is not the subject matter of the claimed invention.

53. By way of example, I have been informed that Booth includes "an easy way to validate a character string used as a date," and/or check for "correctly formatted dates that are not reasonable or even possible," and/or to select a "'safety' date which no [date being entered] can precede [, or] maximum allowable date." (Booth, at 526-28) In addition Booth describes "date manipulation capabilities." Clipper "provides three settings which control the display of dates," [e.g.,] "whether or not the year portion of a date is display [sic] with four digits (including the century) or two digits (not including the century)," [and] "different date display formats." (*Id.* at 939-40)

54. Booth also describes a "SET EPOCH command" which, in the same way as Ohms, "informs the system how to handle date data entry that use only two digits for the year." According to this function, "[w]hen a two-digit year is entered into a date, its year digits are compared with the year digits of the epoch setting to determine the century to place the date into. If the two digits are prior to the setting of SET EPOCH, the year is assumed to be in the next century. If the digits are greater than or equal to the SET EPOCH setting, the year is assumed to be in the current century." (*Id.*, at 941)

55. This is virtually identical to the utilization of windowing to enter dates into a database (where they are then stored in Lilian format) that is disclosed in the Ohms article, as discussed by the Applicant as noted below. In the SET EPOCH function disclosed in Booth, the pivot year defaults to 1900, which "forces any date entered to be considered a

date in the twentieth century." This also means that in this mode, no Y2K date ambiguity problem is recognized or accommodated, even in date data entry.

56. Booth also discloses a function DTOC(), i.e., "[t]he date to character function [which] takes a date variable ... and returns a string representation of the date. The string is recreated in the format specified by the SET DATE or the SET DATEFORMAT command. If SET DATE has not been specified, the default date format is mm/dd/yy." (*Id.*, at 944) Also disclosed is a function DTOS() "date to string function" which "takes a date variable ... and returns a string in the format YYYYMMDD ... ." The formats available for SET DATE are set forth on page 940 of Booth.

57. There is, therefore, no teaching or suggestion in Booth, or in the combination of Shaughnessy, Hazama and Booth, of:

> providing a database with symbolic representations of dates stored therein according to a format wherein M_1M_2 is the numerical month designator, D_1D_2 is the numerical day designator, and Y_1Y_2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time."

Booth's "method of processing symbolic representations of dates stored in a database" teaches utilizing a database with the symbolic representations of dates stored therein in the form of unique integers or numbers, each representative of a unique day or other more granular moment in time, or a combination of a unique day and a number representing a unique time on that day. This is not storage in a M_1M_2, D_1D_2, Y_1Y_2 format. In addition, there is no teaching or suggestion that those dates all fall within a 10-decade period of time. So far as Booth teaches the dates stored in the database can be any span of dates capable of being represented over the span of time capable of being represented by the particular integer date system being used.[5]

58. There is also no teaching or suggestion in Booth, or in the combination of Shaughnessy, Hazama and Booth, of:

[5] Booth, at 99 ("Clipper supports all dates from January 1, 100AD through December 31, 2999.")

selecting a 10-decade window with a Y_A Y_B value for the first decade of the window, Y_A Y_B being no later than the earliest Y_1 Y_2 year designator in the database; ... .

Booth selects, e.g., "nyear" in order to "handle dates that use only two digits for the year [w]hen a two-digit year is entered into a date [by comparing] its year digits ... with the year digits of the epoch setting to determine the century ... ," (Id., at 941). There is no teaching or suggestion of any consideration of "the earliest Y_1 Y_2 year designator in the database."

59. There is also, therefore, no teaching or suggestion in Booth or the combination of Booth with the references as applied by the Examiner, of:

determining a century designator C_1 C_2 for each symbolic representation of a date in the database, C_1 C_2 having ... ;

There is no need to determine a century designator for each symbolic representation of a date in Booth's database, since each is already stored with the century designator included in the date datum so stored in integer format. In addition, the teaching of Booth is to determine a century designator on an individual date datum basis for date data entry, date display, incrementally determining a date based upon a given initial date datum, etc. This calling of certain functions disclosed by Booth to, for example, display a date, or compare two dates, or increment a date from a starting date, are virtually identical to the pertinent disclosure in Shaughnessy, discussed by the patent owner as noted above. As I have been informed has been noted by the patent owner before, Shaughnessy was properly considered by the original Examiner not to have been relevant to the patentability of the claimed invention.

60. There is accordingly also, therefore, no teaching or suggestion of:

reformatting the symbolic representation of the date with the values C_1 C_2, Y_1 Y_2, M_1 M_2 , and D_1 D_2 to facilitate further processing of the dates.

Booth, like Ohms, does not need to do the recited reformatting, since the dates stored in the database in their original format already contain all the information needed to determine the four digit designation of the date, including the century of the particular

date datum. The process of the claimed invention is not needed for dates stored with the century designator already known from what is stored and the Y2K ambiguity not present. Furthermore, the teaching of Booth, like Shaughnessy, is to reformat one or two dates at a time in a called Clipper date functionality and the return to the program from the called subroutine with information resulting from the performance of the programming functionality, e.g., an input to a display, a result of a comparison, a newly calculated date, etc. Booth does not teach facilitating "further processing of the dates" by "reformatting the symbolic representation of the date" "for each symbolic representation of a date in the database."

61. Accordingly, there is also no teaching or suggestion of "sorting the symbolic representations of dates," as recited in claim 4. These are the reformatted symbolic representations. Whatever sorting Booth teaches does not need to first reformat the date data, since the integer format can be and is sorted in its initial format. The method of the claimed invention, including the reformatting steps is simply not relevant to a database that stores date data as Clipper does, in integer format, as described in Booth.

62. There is accordingly also no teaching or suggestion of "reformatting each symbolic representation of a date into the format C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2," as recited in claim 5. Neither is there a disclosure of "sorting the symbolic representations of dates using a numerical-order sort," as recited in claim 6. There is no disclosure of "storing the symbolic representation of dates and their associated information back into the database," as recited in claim 9 nor "manipulating information in the database having the reformatted date information therein," as recited in claim 10.

63. In addition, accordingly, there is no teaching or suggestion of "converting pre-existing date information [within a database] having a different format into the format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator and Y_1 Y_2 is the numerical year designator," as recited in claim 7. This process step is recited as part of the "step of providing the database" upon which the subsequent process steps recited in the claimed invention, are carried out. The fact that Booth, or other art,

teaches converting date data from one format into the recited format, does not teach it as part of the process of the claimed invention. Similarly, there is no teaching or suggestion in Booth of "selecting Y_A Y_B such that Y_B is 0 (zero)," as recited in claim 8, even though SET EPOCH can and does use pivot years ending in 0. SET EPOCH, as noted above, is not a process according to the claimed invention.

64. There is no teaching or suggestion in Ohms or Booth of:

> providing a database with symbolic representations of dates stored therein according to a format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator, and Y_1 Y_2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time; ...

Unlike this recitation of claim 1, Ohms and Booth teach providing a database with the dates in a Lilian or integer format.

65. There is accordingly also, therefore, no teaching or suggestion of:

> providing a database ... all of the symbolic representations of dates falling within a 10-decade period of time; ... .

Ohms and Booth teach having data in the database in Lilian or integer format, i.e., in the former case within a ninety-nine million day window (seven chronological day date numbers starting at a given date).

66. There is accordingly also, therefore, no teaching or suggestion of:

> selecting a 10-decade window with a Y_A Y_B value for the first decade of the window, Y_A Y_B being no later than the earliest Y_1 Y_2 year designator in the database; ... .

Ohms and Booth teach or suggest selecting a Y_AY_B for the first decade based upon dates that are being input into the database.

67. There is accordingly also, therefore, no teaching or suggestion of:

determining a century designator C_1 C_2 for each symbolic representation of a date in the database, C_1 C_2 having ... ; ... .

Ohms teaches entering date data into the database to be converted into Lilian format for storage and manipulation within the database. He does not teach or suggest determining a century designator for data in the database. Lilian format needs none.

68. There is accordingly also, therefore, no teaching or suggestion of:

reformatting the symbolic representation of the date with the values C_1 C_2, Y_1 Y_2, M_1 M_2 , and D_1 D_2 to facilitate further processing of the dates.

Ohms teaches reformatting into Lilian format for purposes of facilitating the later processing of the date data in the database utilizing the Lilian format.

69. Ohms would not have made the claimed invention as recited in claim 1 obvious to a person of ordinary skill in the art at the time the invention was made, under 35 U.S.C. §103. As explained above, not only does Ohms not teach or suggest the claimed invention recited in claim 1, it clearly teaches away from virtually every step of the method of the claimed invention as recited in claim 1. The same can be said of Booth.

70. The combination of Ohms with Hazama, or Ohms with Hazama and Booth would not have made the claimed invention as recited in claim 1 obvious to a person of ordinary skill in the art at the time the invention was made, under 35 U.S.C. §103. As explained above, not only does Ohms not teach or suggest the claimed invention recited in claim 1, it clearly teaches away from virtually every step of the method of the claimed invention as recited in claim 1. The combination of these references would not result in there being present all of the elements of the claim as recited in claim 1 of the Dickens patent as issued. The same can be said of Booth.

71. For like reasons accordingly there is also no teaching or suggestion of:

sorting the symbolic representations of dates (claim 4); or reformatting each symbolic representation of a date into the format C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2 (claim 5); or sorting the symbolic representations of dates using a numerical-order sort

(claim 6); or storing the symbolic representation of dates and their associated information back into the database (claim 9).

72. There is also accordingly no teaching or suggestion of:

The method of claim 9, including the additional step, after the step of reformatting, of

manipulating information in the database having the reformatted date information therein (claim 10).

73. In addition, accordingly, there is no teaching or suggestion of:

converting pre-existing date information having a different format into the format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator and Y_1 Y_2 is the numerical year designator (claim 7) or selecting Y_A Y_B such that Y_B is 0 (zero) (claim 8).

74. The above is also applicable to claims 11- 15 in the Dickens patent as originally issued.

75. The above discussion as to claims 1-3, 5, 7, 9 and 11-12 is also applicable to claims 16-18, 20, 22, 24-25 added in the Reissue application.

76. The above discussion of claims 2, 4 and 6, is also applicable to claims.19, 21, 23 added in the Reissue application.

77. Neither Shaughnessy, nor Ohm, nor Booth nor Hazawa, nor any combination of these references discloses or suggests the claimed invention as recited in Claim 16. At a minimum, as noted above, these references separately or collectively fail to teach or suggest "reformatting the symbolic representation of each symbolic representation of a date in the database ... ." Further, at a minimum, they do not teach or suggest doing the reformatting "without the addition of any new data field to the database ... ." In addition there is no teaching or suggestion of "the reformatted representation of each date in the

database having the values C_1 C_2, ... , in order to facilitate collectively further processing the reformatted symbolic representations ... of each of the dates."

78. Ohms and Booth, utilizing Lillian and integer date formats, respectively, do not reformat dates in the database at all, and do not even have the Y2K ambiguity problem addressed by the present invention. Shaughnessy and Hazama encounter a date datum and call a subroutine or module to process the single date, or at most two dates, for resolution of the Y2K ambiguity problem. This is not the claimed invention, as distinguished from these references by at least the recitations noted above in regard to the originally issued claims and, e.g., in paragraph 67.

79. As to claim 19, neither Shaughnessy, nor Ohm, nor Booth nor Hazawa, nor any combination of these references discloses or suggests the claimed invention as recited in Claim 19. For the reasons noted above, the "symbolic representations of [each of the] dates [in the database]" is not taught to be produced, and/or is not taught to be produced according to the method of the claimed invention, as recited in claim 16. Therefore, whatever sorting is done in these references is not the claimed sorting.

80. The same is true with respect to claim 20, as was the case with claim 19, as noted above. In addition the references do not teach or suggest "reformatting each symbolic representation of a date ... separately from the symbolic representations in the database."

81. With respect to claim 22, to the extent that the references or any of them discloses or suggests "converting pre-existing date information ... " as recited in claim 22, they do not teach or suggest doing so as a part of the method recited in the allowable claim 16.

82. With respect to claim 23, the same can be said as with respect to claim 22 above.

83. With respect to claim 24, neither Ohms, Shaughnessy, Booth nor Hazama, nor any combination of these references teach or suggest "storing the symbolic representation of dates and their associated information back into the database." Ohms and Booth do not

store symbolic representations of dates, as recited, but instead store dates in, respectively, Lilian or integer date formats. Shaughnessy does not send any reformatted date data for storage anywhere outside at most the subroutine/module called to handle one or two date representations at any given execution of the subroutine/module. The same is true for Hazama as understood from the translation. Even so at best, Hazama, discloses returning a modified data datum, which is not in accordance with the process of the Dickens invention, wherein the date data field in the legacy database is not to be modified. The claim indicates that the symbolic representations of the date along with its "associated information," i.e., the other datum fields can be returned, e.g., after sorting or other manipulating, e.g., so that the data in the database will be, e.g., in the newly sorted order.

84. With respect to claim 25, neither Ohms, Shaughnessy, Booth nor Hazama, nor any combination of these references, teach or suggest "manipulating information in the database having the reformatted date information therein." Ohms and Booth do not manipulate reformatted symbolic representations of dates, as recited, but instead manipulate dates in, respectively, Lilian or integer date formats. Shaughnessy and Hazama do not manipulate any reformatted date information in the database.

85. The above discussion regarding the corresponding claims also applies to claims 26-30.

86. The above discussion also applies to claims 31-33.

87. With respect to claims 31-33, neither Shaughnessy, nor Ohm, nor Booth nor Hazama, nor any combination of these references, discloses or suggests the claimed invention as recited in Claim 31. At a minimum, these references separately or collectively fail to teach or suggest "reformatting the symbolic representation of each symbolic representation of a date in the database ... ." Further, they do not teach or suggest doing the reformatting "without the addition of any new data field to the database ... ." In addition there is no teaching or suggestion of "the reformatted representation of each date

in the database having the values C_1 C_2, ... , in order to facilitate collectively further processing the reformatted symbolic representations ... of each of the dates."

88. Ohms and Booth, utilizing Lillian and integer date formats, respectively, do not reformat dates in the database at all, and do not even have the Y2K ambiguity problem addressed by the claimed invention. Similarly the discussion of Shaughnessy and Ohms in this regard noted above is applicable here.

89. The above discussion of the references is also applicable to claims 34-59.

90. With respect to claim 34, neither Shaughnessy, nor Ohms, nor Booth nor Hazawa, nor any combination of these references, discloses or suggests the claimed invention as recited in Claim 34. At a minimum, these references separately or collectively fail to teach or suggest "converting each of the symbolic representations of dates stored in ... the database ... ." Further, they do not teach or suggest doing the reformatting "by windowing ... each of the respective dates as stored ... without the addition of any new data field to the database ... ." In addition there is no teaching or suggestion of "running a program collectively on each of the converted symbolic representations of each of the respective dates ... ."

91. With respect to claim 35, in addition to the discussion above with respect to claim 34, these references do not separately or collectively teach or suggest "opening the database prior to the step of converting" in the process as recited in claim 34.

92. Similarly with respect to claims 36 and 37, in addition to the discussion above with respect to claims 34 and 35, these references do not separately or collectively teach or suggest "collectively sorting the converted symbolic representations prior to the step of running the program" in the processes as recited in claims 34 and 35.

93. Similarly with respect to claims 38 and 39, in addition to the discussion above with respect to claims 34 and 35, these references do not separately or collectively teach or

suggest "collectively manipulating the converted symbolic representations prior to the step of running the program" in the processes as recited in claims 34 and 35.

94. Similarly with respect to claims 40 and 41, in addition to the discussion above with respect to claims 34 and 35, these references do not separately or collectively teach or suggest "collectively sorting the converted symbolic representations according to a different data field contained in the database ... ." Neither do they teach or suggest doing this "prior to the step of running the program on the converted symbolic representations" as recited in claims 40 and 41.

95. Similarly with respect to claims 42 and 43, in addition to the discussion above with respect to claims 34 and 35, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations according to a different data field contained in the database ... ." Neither do they teach or suggest doing this "prior to the step of running the program on the converted symbolic representations" as recited in claims 42 and 43.

96. Similarly with respect to claims 44 and 45, in addition to the discussion above with respect to claims 34 and 35, these references do not separately or collectively teach or suggest the process as claimed in claims 34 and 35 wherein in addition "the program performs an operation which manipulates the data in a data field associated with the at least one date data field of the database according to the converted symbolic representation of the date" as recited in claims 44 and 45.

97. Similarly with respect to claims 46 and 47, in addition to the discussion above made with respect to claims 34 and 35, these references do not separately or collectively teach or suggest the process as claimed in claims 34 and 35 wherein in addition "the step of converting includes converting at least a substantial portion of each of the plurality of symbolic representations of dates ... and repeating this step until each of the date data entries in the at least one date data field is converted" as recited in claims 46 and 47.

98. Similarly with respect to claims 48 and 49, in addition to the discussion above with respect to claims 46 and 47, these references do not separately or collectively teach or suggest "collectively sorting the converted symbolic representations prior to the step of running the program" as recited in claims 48 and 49.

99. Similarly with respect to claims 50 and 51, in addition to the discussion above with respect to claims 46 and 47, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations" as recited in claims 50 and 51.

100. Similarly with respect to claims 52 and 53, in addition to the discussion above with respect to claims 46 and 47, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations according to a different data field ... prior to the step of running the program" as recited in claims 52 and 53.

101. Similarly with respect to claims 54 (as amended) and 55, in addition to the discussion above made with respect to claims 52 and 53, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations" as recited in claims 54 and 55.

102. Similarly with respect to claims 56 - 59, in addition to the discussion above with respect to claims 52 - 55, these references do not separately or collectively teach or suggest a process "wherein the program performs an operation which manipulates the data in the date data field ... according to the converted symbolic representation of the date" as recited in claims 56 -59.

103. The above discussion of the references applies as well to claims 60-65.

104. With respect to claim 60, neither Ohms, Shaughnessy, Booth nor Hazama, separately or collectively teaches or suggests "converting each of the symbolic

representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity," as recited in claim 60. Or at a minimum they do not also teach or suggest doing this "without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting," as recited in claim 60. Neither do they separately or collectively teach or suggest "running a program on each of the converted symbolic representations of each of the respective dates ...according to the dates represented by the converted symbolic representations," as recited in claim 60. In addition, also at a minimum, they do not separately or collectively teach or suggest doing so "separately from the date data symbolic representations of dates contained in the at least one date field," as recited in claim 60.

105. The discussion in paragraph 94 applies also to claim 61.

106. With respect to claim 62, the same discussion above regarding claim 60 also applies to claim 62 and in addition, these references do not teach or suggest the step of "converting" including "without the addition of any new data field to the database for purposes of such windowing and converting," as recited in claim 62. Neither do they separately or collectively teach or suggest "storing the converted symbolic representations separate from the at least one date field of the database," as recited in claim 62. Neither do they teach or suggest "running a program on the stored converted symbolic representations," as claimed in claim 62.

107. With respect to claim 63, the same discussion above as to claim 62 also applies to claim 63.

108. With respect to claims 64 and 65, the same discussion above as to claims 62 and 63 applies to claims 64 and 65 with the exception that the claimed step of "converting" includes "without modifying any of the symbolic representations of date in the at least one date field of the database for purposes of such windowing and converting," which is not taught or suggested by these references separately or collectively in a process as

defined by the recitations of claims 64 and 65, nor with the additional step of "storing the converted symbolic representations separate from the at least one date field in the database," as recited in claims 64 and 65.

109. The same discussion of the references above applies as well to claims 66-69.

110. With respect to claims 66 and 67, the references do not teach or suggest, separately or collectively "reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new date field to the database ...; and repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations," as recited in claims 66 and 67.

111. With respect to claim 68, these references do not teach or suggest, separately or collectively, "reformatting the symbolic representation of each symbolic representation of a date in at least one date field in the database, without the addition of any new date field to the database ... in order to facilitate processing of the reformatted symbolic representations ... by running a program on the reformatted symbolic representations of each of the dates" as recited in claim 68.

112. With respect to claim 69, the same discussion above with respect to claim 68 applies to claim 69 an in addition, the claim recites "sorting the reformatted symbolic representations (in the recited format) ... and running a program on the reformatted symbolic representations of each of the dates," which is not taught or suggested by these references, separately or collectively.

113. The above discussion of the references also applies to claims 70-76.

114. The same discussion of claims 61-65 apply as well to the claims 70-76 except that the claims 70 and 71 recite both "without the addition of any new data field" and

"without modifying any of the symbolic representations of dates in the at least one date field ... ."

115. The description contained in the Specification of the Dickens patent as filed, is clear, concise and descriptive of the claimed invention and enabling of the claimed invention, especially of the claims as filed, but also including those claims added in the Reissue Application. This includes the Specification as filed with or without the Exhibit A filed with the application but not printed with the patent as issued, but it is even more clearly present with the disclosure of Exhibit A.

116. It is plain from the discussion in the Dickens Patent Specification referenced in paragraphs 6-8 above that dates stored in a database in pure numerical form, e.g., Lilian, as in Olms, or in other form, e.g., binary form, e.g., integer form as in Booth, with a unique number representing each day (or each minute or second or part thereof, depending on the granularity) in a chronological sequence of days (minutes, seconds, etc.) from a particular starting date, as in Ohms or Booth, is not even analogous art to the present invention. In those databases, the number stored includes by definition the year in four digits, and is not susceptible to the problems solved by the invention claimed in the Dickens patent as issued and/or as added in the Reissue application, among others, being that a sort on dates stored as discussed above in the cited portions of the Dickens patent will be subject to, e.g., the system being "unable to distinguish between the year 2000 and the year 1900, for example, the latter is also represented by the two digit code 00." (See Shaughnessy, Col. 1, lines 23-25).

117. Therefore, both Ohms and Booth add nothing to the disclosures of Shaughnessy and/or Hazawa related to teaching or suggesting the invention as claimed in the Dickens patent as issued and/or as added in the Reissue application. Ohms and Booth teach a utilization of windowing, but not in the context of any suggested solution to the problem that is solved by the invention of the Dickens patent. In fact they teach away, since they teach the storage of the dates in a form that is not the "symbolic representation[]" dealt with in the Dickens patent and is not subject to the system being "unable to distinguish"

dates stored in that fashion. Shaughnessy and Hazawa already teach a utilization of windowing and the added teachings of either or both of Ohms and Booth, beyond using windowing for something not related to what windowing is used for in the Dickens patent, are irrelevant to the invention of the Dickens patent.

118. One of ordinary skill in the art would readily understand the Specification of the Dickens patent, e.g., as mentioned in paragraph 9 above, to reflect the issue that caused the Y2K problem in the first place. Databases (referred to herein generally as "legacy databases") created initially during a period of time when memory was relatively very much more expensive than it is today or even has been for the last decade or so, were designed to have as small a data field for each needed item of information stored in the database as possible. For the year portions of dates stored, e.g., in a single date data field (e.g., MMDDYY) or a set of data fields (e.g., a separate MM, DD, and YY data field), as is the case in the databases relevant to the Dickens patent, this meant two character year date information with no century designation.

119. The just referenced concern is at least part of the reason, as would have been well understood by one skilled in the art, that other representations, e.g., that are not symbolic, as that term is defined in the Dickens patent, were created, e.g., Lilian as in Ohms and integer as in Booth. Ohms and Booth represent ways to initially store a complete date having day month and year to four digits, and maybe even with less bytes than would be required for a symbolic representation in the form, e.g., DD/MM/Y, not to mention DD/MM/YYYY.

120. Therefore, existing legacy databases, which have Lilian or integer date data formats are not subject to a Y2K ambiguity problem. Legacy databases in which the dates are stored in a date data format including a data field or set of data fields, including only the storage of two year date designation characters, do have, as the Dickens, Shaughnessy and Hazama patents recognize, the Y2K ambiguity problem.

121. While an existing legacy data base having the Y2K ambiguity problem may be modified to place the date data in the database in another format, e.g., containing YYYY instead of just YY, or into Lilian or integer format, or by adding to the legacy database another field, e.g., a CC, century designator field, as noted by both the Dickens patent and Shaughnessy, this can be complex, costly and maybe not effective.

122. Replacing the one date data field or set of date data fields in a legacy database of the type described by the Dickens patent to suffer the Y2K date ambiguity problem, as would have been understood by those skilled in the art at the time of the filing of the application leading to the Dickens patent, may be simply unworkable or at best costly, time consuming and subject to numerous errors that may be even more costly and expensive to be fixed, if they even can be, after such a conversion. In short the best solution available in the prior art, as suggested by both Shaughnessy and the Dickens patent, may have been, as many people did, abandon the legacy database in favor of a totally new one in which the years date data is stored unambiguously, e.g., as YYYY. If such abandonment was not feasible, then there existed also the possibility of creating an entirely new database and transferring all of the data entries in all of the fields into the newly created database, if possible without massive clerical error. Neither solution was really economically viable as suggested in Both Shaughnessy and the Dickens patent. Shaughnessy proposed a solution to this problem but his is significantly different from that of the claimed invention in the Dickens patent, and also not nearly as effective.

123. One of ordinary skill in the art at the time of the filing of the application that led to the Dickens patent would have understood the many problems in modifying a legacy database containing only room for date data in the format YY to a modified format, e.g., containing a YYYY date data format, some of which, by way of example only, might include:

(1) a database designed to be organized and contained in memory in a certain way, e.g., to conserve space or improve accessibility or both may physically (electronically) not be susceptible to expanding the YY date characterization into, e.g., YYYY; or

(2) pointers and other links, e.g., between data fields included in a string of date data information, i.e., DD, MM, YY, or other such links, e.g., between such a string entered in the database and another string that the database used, e.g., in comparing, or, e.g., for date sorting purposes, or, e.g., the initial data entry system of the database, or many other reasons that would have been understood by those skilled in the art at the time of the filing of the application leading to the Dickens patent, may be set up to look for only a specified memory location or locations which may not be sufficient to contain the new year date data in, e.g., the expanded format.

124. This is the problem addressed in the Dickens patent and its disclosure, e.g., as referenced above, e.g., in paragraphs 10 - 13, as would be done by one of ordinary skill in the art, must be construed with that fact in mind.

125. One of ordinary skill in the art would necessarily have understood from the disclosure of the Dickens patent, e.g., as referenced in paragraphs 10 – 11, with or without Exhibit A, that the steps of determining a century designator and reformatting each of the dates in the database is to be done without requiring additional or modified date data fields in the existing legacy database. The disclosure specifically says that avoiding having to do such a modification of the existing legacy database is the very reason for the claimed invention. This is also true in light of the claimed purpose being to "facilitat[e] further processing of the dates," and because of the disclosure in the Dickens patent discussed further below that "[o]nce the symbolic representations of the dates are reformatted ... the date information may be sorted ... ."

126. It is plain from the disclosure of the Dickens patent, with or without Exhibit A, e.g., as referenced above in paragraphs 12 and 13, and would have been so understood by one of ordinary skill in the art, based upon the context of the disclosure as a whole of the Dickens patent, with or without Exhibit A, at the time of filing, that what is disclosed is the conversion, wholesale, of the dates in the database and running some program on the large number of dates so converted and that this is done without changing the underlying data fields in the legacy database from which the date data information was originally

obtained and to which it may be returned. Similarly, it is plain, and would have been so understood by one of ordinary skill in the art, that to perform the method of the Dickens patent, the converted dates must be stored somewhere outside of the existing database date data fields, otherwise sorting and other manipulations by applications programs could not be done on all of the dates taken from the database and reformatted according to the claimed invention.

127. Shaughnessy discloses the receiving from the processor into the subroutine of the one or two dates taken from the legacy database date data field(s) that Shaughnessy's subroutine operate upon, without specifically disclosing where outside the original date data field in the legacy data base this storage would occur. Similarly, those skilled in the art would have understood that all of the converted dates from the legacy database, according to the invention as claimed in the Dickens patent, ["each symbolic representation of a date in the database," and "reformatting the symbolic representation of the date [for each such date]" would require some memory and it would not be the date data fields of the legacy database. Only in this manner, as would also have been understood by those of ordinary skill in the art from the disclosure in the Dickens patent, with or without Exhibit A, could a program be run on the dates so reformatted to accomplish the subsequent claimed steps of, e.g., "sorting" and "manipulating."

128. It is clear from the prosecution history of the Dickens patent before it was originally issued, specifically as referenced above in paragraphs 14 – 17, that when applicant's counsel said that the claimed method of the patent application "should accept dates from data storage ..., " applicant's counsel was referring to data storage where the database, i.e., a legacy database, was stored with only the availability of Y_1Y_2, to discern the proper century. In addition it is clear that when applicant's counsel again uses this term in stating "[h]owever the method of the present invention need not store the converted date in data storage," that this is the same data storage where the original legacy database is stored, with its limitations, e.g., as to ability to store date data in other than a two character year date data format. Applicant's counsel immediately goes on to state "[i]nstead, the original dates in the data storage remain undisturbed." Thus the original

legacy database remains as it was with no changes. Applicant's counsel then immediately thereafter goes on to say that "[t]his aspect of the present invention thus allows conversion of dates to compensate for century designations without requiring the addition of data fields to permanently store the century designations." This also makes in clear that when applicant's counsel asserted that "the claimed invention does not require that the converted data that includes the century designations be stored in data storage," he was referring to storing the converted data back in the original date data fields of the legacy database, within the "data storage." It is evident also that the statement by applicant's counsel that "the amended set of claims does not require storage of the converted data and therefore imposes no requirement for new data fields," is referring to storage of the converted data in new data fields in the legacy database, within the "data storage." Further it is clear that the comment of applicant's counsel, that "conventional date formatting systems typically require additional data fields for storage to accommodate the century designations," means that the problem being addressed is being able to sort, manipulate and otherwise run programs on these date representations without modifying the existing fields in the database or changing that data in those fields permanently.

129. As the specification of the Dickens patent points out and as those in the art would have known at the time, the legacy database is not readily susceptible of changing the format of those date data fields for permanently modifying the date data format contained in the legacy database. Applicant's counsel's comment that "[t]hese additional data fields are necessary because conventional systems disclose a permanent reformatting of stored data," clearly refers the undesirable prior art solution of the Y2K problem by reformatting the legacy database itself and then, e.g., reformatting all of the date data and also the fact that this solution does require that new data fields be added to the legacy database.

130. Additionally, in the context of this argument, the applicant's counsel's assertion that "[t]he claimed invention, on the other hand, does not require that the reformatted data be permanently stored. Instead, the method of [the] claimed invention encompasses

embodiments in which the date information is initially reformatted and converted to have century designations, but does not require that the reformatted dates be stored," is plainly talking about the lack of need for permanent storage within the date data fields of the legacy database. One skilled in the art would plainly understand this from the disclosure, with or without Exhibit A.

131. Even Shaughnessy, as noted above, in his method has to store one or two dates processed each time by the called subroutine in some form of data storage, even if it is a register or cache memory in the processor. Clearly the set of date data, which has been converted and reformatted, according to the claimed invention, must be stored somewhere for the additional process steps of, e.g., sorting or manipulating (or as shown in Exhibit A, sorting by model number and then sorting by date) in order for these later process steps to be performed. This would have been understood by one of ordinary skill in the art from the disclosure, especially with Exhibit A. What is not changed is the date data stored in the data storage that contains the original legacy database. Not only would this have been evident to one of ordinary skill in the art at the time of the filing of the Dickens patent, this is what applicant's counsel argued to the Examiner:

> As stated previously, the method of one embodiment of the claimed invention reads the dates from the database and temporarily reformats the dates with century designations. Data manipulation programs are then performed on these reformatted dates, such as sorting the dates. However, once the data manipulations are complete, the reformatted dates need not be stored in data storage. (Emphasis added)

132. This is the same "data storage" referenced by applicant's counsel to be where the legacy database was stored. Applicant's counsel continued:

> [i]nstead the dates in the data storage can remain the same as they were prior to the temporary reformatting of the data by the method of the claimed invention. Thus in these embodiments, the method of the claimed invention does not require additional data fields for storage because the reformatted dates with the century

designations are only used 'on the fly' for data manipulation and are not stored in data storage.

133. It should be plain from the above referenced prosecution history that the original Examiner understood the claimed invention to be what applicant now asserts it is according to the meaning of the claims as allowed in the Dickens patent. This meaning applies as well to the claims added herein in the Reissue application. That is, the method allows the extraction from an existing legacy database with date data stored in a format, e.g., using only Y_1Y_2, that is Y2K ambiguous, temporarily converting and reformatting each of the extracted dates to a format, e.g., C_1C_2,Y_1Y_2, that is not Y2K ambiguous, performing data manipulation programs on these reformatted dates, that are not stored in the original legacy database fields, but necessarily must be stored somewhere separate from or outside of the original legacy database fields, and utilizing the results of the data manipulation program, without having to have modified the original legacy database and its original fields, formats, links, etc.

134. Otherwise, the original Examiner would not have removed the rejection based on the change in the claims from "reformatting the symbolic representation of the date in the database" or "reformatting each date in the database," to, respectively, to "reformatting the symbolic representation of the date ... to facilitate further processing of the dates" and to facilitate further "reformatting each date ... to facilitate further processing of the dates," as was done in the Supplemental Response of April 2, 1998, which resulted in allowance. The original Examiner stated:

> The Prior Art of Record ... does not anticipate nor suggest the set of limitations of the claims, comprising the threshold year digits as used to determine a pair of century digits to be used for computation, but without enlarging the number of date digits in of the database.
>
> Further stated the Original Examiner in an Interview summary of April 2, 1998:
>
> It was agreed that the summary of the invention, and the arguments of the response, were not entirely in conformity with the claims, which would be potentially allowable if the use of additional century digits did not include their storage in the database.

135. Claim 10, as noted above was amended to clarify the same conflict with the disclosure as the original Examiner recognized in the claims as originally filed, i.e., that the reformatted dates are not stored back into the database.

136. Given the interpretation of the claims as filed originally and issued as amended in the original application leading to the Dickens patent, and the claims added in the current Reissue application that clarify further the meaning of those original claims, clearly the same reasons as asserted by the original Examiner for allowance of the claims in the original application apply to claims 1-76 in the present application.

137. It is not correct, as the Examiner asserts, that Shaughnessy "teaches modifying those dates that have a two digit identifier less than some predetermined pivot date, changing the format of the date, and sorting the results." To the extent that "those dates" is intended by the Examiner to mean the recited "all of the symbolic representations of dates" and/or "each symbolic representation of dates," Shaughnessy's teaching, at best, is "modifying those dates that have a two digit identifier less than some predetermined pivot date, changing the format of the date" only with respect to one, or at most two, dates sent to a called subroutine when an application encounters a two digit date data and an instruction, e.g., to determine if that date is in the past or future, i.e. to compare it to some other single fixed date, or, e.g., to compare two dates encountered by the application program.

138. Shaughnessy does not perform the step of "sorting the results," if the Examiner means the claimed results of "reformatting" "each" or "all" reformatted symbolic representations in the data base. Shaughnessy sorts between a single fixed date and a forwarded date or between two dates forwarded from the application and returns to the application a "parameter" indicating the result of, e.g., the sorting of the two dates.

138. Hazama, similar to Shaughnessy, has a "computer system ... processing section [which] replaces the code for the tens place in the last two digits of the year AD with a

code that maintains the year order." To do this, the two digit date code is sent by the program processor, referred to as "work area" 8 or "clear area" 8, to a module 10 and the modified date is returned to the processor "work area" or "clear area" 8. Therefore, even with the disclosure in Hazama "of the need for the pivot date to be less than any date in the database" the claimed invention is no more disclosed than in Shaughnessy.

139. It is not correct that Shaughnessy teaches or suggests the "process of converting *all dates* in the database, wherein two digit dates are converted into four digit dates as taught in Shaughnessy ... ." (Emphasis added)

140. It also does not follow from the asserted combination of Shaughnessy and Hazama as proposed by the Examiner because "it follows that one of ordinary skill in the art of programming would know and would be adept at setting parameters to correctly process a set of data," means that the combination of Shaughnessy and Hazama results in the claimed invention. Assuming that the Examiner means "sets of data" to be the recited "all of the symbolic representations of dates" and/or "each symbolic representation of dates" in the database, since Shaughnessy, as noted above, contains no such disclosure. Because of the way Shaughnessy is disclosed to operate, it cannot perform the claimed process, and therefore, teaches away. Shaughnessy by returning a "parameter" to the program cannot reformat each or all of the date data representations in the data base and then perform further programming "sorting" or "manipulating" on the reformatted date data, since the "parameter" returned to the program is specific to an operation, e.g., comparison, specific only to the two particular date representations being operated on by Shaughnessy for purposes of returning the parameter to the program. The "parameter" simply indicates, e.g., the one date is greater than, equal to or less than the other, and is not correlated to any other date data representation in or extracted from the database for purposes of further processing. It is, therefore, not a conversion and reformatting that "facilitates further processing of the dates" taken from the database as claimed.

141. The Examiner's comment about Shaughnessy complemented by "logical necessity" disclosing the entire claimed invention is not correct in light of fact that Shaughnessy,

with the addition of the selection of a pivot date based on the earliest date in the data base, still does not result in each and every element of the claimed invention.

142. Shaughnessy does not "suggest[] the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq)." In the cited passage, Shaughnessy is either saying that as to, e.g., a legacy data base it is difficult and expensive to modify the format, fields, etc. for the data base to only include 4 character date representations, with which applicant is in agreement (see Col. 1, lines 35 – 41 of the Dickens Patent), or teaching away from the applicant's proposed solution, or both, as the cited passage is not entirely clear as to what would be the objectionable thing to avoid. See also Shaughnessy's discussion in Col. 4, lines 7-26. The fact that Shaughnessy specifically teaches a separate way than the claimed invention to solve this problem, also teaches away from the claimed invention. At best this cited portion of Shaughnessy recognizes what the problem is that both Shaughnessy and the claimed invention set out to alleviate, but Shaughnessy takes a different approach.

143. Shaughnessy does not "discloses the claimed 'all of the symbolic representations of dates falling within a 10 decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1= THE DATE CYCLE IS 100 YEARS)." The table of the Appendix to which the Examiner refers is identified by Shaughnessy as "illustrating a sample of the *types of date formats* the present invention can support." (Col 3, lines 43-45. The fact that a cycle for dates of a given one of may listed formats in the referenced table may be 100 years teaches nothing about the selection of a range of dates that are *in a data base* upon which the method of Shaughnessy is utilized. Indeed, a number of the ranges listed as to which Shaughnessy says "the present invention can support," are longer than 100 years. Once again the teaching of Shaughnessy is away from the present invention, in that it does not require claimed recitation "all of the symbolic representations of dates falling within a 10 decade period of time."

144. For this same reason, it is inappropriate hindsight analysis using only the teaching of the applicant's disclosure to say that:

As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, incorporate this limitation.

145. The "commercial or industrial" databases which Shaughnessy invention "can support," are specifically identified by Shaughnessy to include data bases where the data contained could include a span of over 100 years

146. Shaughnessy does not teach:

the step of 'determining a century designator C. sub.1 C. sub.2 for each symbolic representation of a date in the database, C. sub. 1 C. sub.2 having a first value if Y. sub.1 Y. sub. 2 is less than Y. sub. A Y. sub. B and having a second value if Y. sub. 1 Y. sub. 2 is equal to or greater than Y. sub. A Y. sub. B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq).

While this portion of Shaughnessy does disclose the form of windowing that Shaughnessy proposes, using the install date as the pivot date, in the context of the remainder of the disclosure of Shaughnessy this is not disclosed to be done for "each symbolic representation of a date in the database," in preparation for and facilitation of further processing of the reformatted symbolic representations. As noted above Shaughnessy does windowing and reformatting for one or two date representations at a time in the called subroutine.

147. The same can be said for the cited portion of Shaughnessy cited to disclose:

the comparison of the YYMMDD portion of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq).

148. Shaughnessy also does not disclose:

the step of 'reformatting the symbolic representation of the date with the values C. sub. 1 C. sub.2, Y. sub. 1 Y. sub.2, M. sub. 1 M. sub. 2 , and D. sub. 1 D. sub. 2 to

facilitate further processing of the dates' by appending the determined century value before the YYMNMD date in order to yield a CCYYMMDD date format (col.5, lines 46-51; col.6, lines 57-65 et seq) ... .

The "reformatting the symbolic representation of the date" must be read in the context of the earlier claim recitations that define the context of this process limitation to mean performing reformatting on "each" or "all" date data representations that have previously been the subject of the claimed form of windowing prior to this reformatting step. Further the claimed reformatting is to "facilitate further processing of the dates." As noted above, Shaughnessy reformats at most two pieces of date data information from the data base at a time and returns a parameter that gives a result of processing done in the called up subroutine of Shaughnessy, but does not "facilitate further processing of the dates."

149. Shaughnessy does not disclose:

> "returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col.1, lines 47-54 et seq).

The cited portion of Shaughnessy, which appears to be in Col. 2 as opposed to Col. 1, refers to "passing at least one date field *to* the subroutine [and] a means for *returning a parameter* to the application ... ." The "parameter," is not a reformatted date or dates, and is "for use by the application program in further operation." (Col. 2, lines 53-54, Emphasis added) An example of this is "[I]f the result received from the subroutine [the parameter] indicates that the date the next payment is due is greater than today's date [the program can go on to] indicate that the account is OK." (Col 4, lines 59-61) Shaughnessy also notes as an alternative to the above, the Shaughnessy process may "pass[] at least one date field which is representative of at least two dates *to* the subroutine, determining which if the two dates corresponds to the date field operation according to a predetermined criteria , performing the date operation on the date field, and *returning the parameter* to the application program ... ." (Col. 2, lines 59 – 64, Emphasis added).

150. Contrary to the examiner's assertion at the time of the filing of the Dickens Patent, it is not correct that:

The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy teaches a very specific way of selecting a one hundred year window, and it is not based on the earliest date in the data base. In this regard Shaughnessy actually teaches away. Further, the import of the latter portion of the Examiner's above quoted view is not understood. The statement regarding "as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall within the 20th century." If the Examiner mean that this is what Shaughnessy would teach one of ordinary skill or that the combination of Shaughnessy and Hazama would teach one of ordinary skill, then all dates falling in the 20th is a situation in which there is no Y2K problem to solve, so that either Shaughnessy or the combination of Shaughnessy and Hazama also teach away. Is this the Examiner's intent?

151. In regard to the rejection of claim 9, the cited portion of Shaughnessy, actually including also Col. 5, lines 10 – Col. 6, line 35 refers to the setting by the subroutine of the "current date (box 14), the end of the hundred year cycle (box 16) and the two possible century values (box 18)" for utilization inside the subroutine and has nothing to do with the recited "'storing the symbolic representation of dates and their associated information back into the database after the step of reformatting'" from claim 9.

152. As to the rejection of claim 9 Shaughnessy also does not disclose "storing the symbolic representations of dates and their associated information back into the database," i.e., after the step of "facilitating the further processing," and after such further processing, e.g., date sorting, the dates and information in Shaughnessy may not be reorganized in the database according, e.g., to the results of the sorting. Further

Shaughnessy does not teach doing so without modifying the symbolic representations of dates in the database itself.

153. Booth is non-analogous art and actually teaches away from the present invention as claimed. Booth and the Clipper system described therein use integer dating which does not suffer from the problems sought to be solved by the claimed invention, i.e., there is no Y2K date ambiguity that needs to be addressed in the processing of date data stored in a database according to the Clipper system. Booth does disclose windowing and using a ten decade window, but it is not in the context of the claimed invention. For example, it is not for the purpose of "facilitating further processing of the dates [in the database]." Booth's use of windowing is also not disclosed to be "reformatting the symbolic representation of the date [for each/all representations of dates stored in the database]." In addition Booth does not teach "sorting the symbolic representations of the dates" in a CCYYMMDD or like format, since dates in Booth are sorted by comparison of the integer value that computes to the appropriate date, including its four character year value.

154. Modifying Shaughnessy or the combination of Shaughnessy and Hazama with Booth, which teaches storing and manipulating (operating programs on) date data that is in integer form, would render Shaughnessy and/or the combination of Shaughnessy and Hazama inoperative for their intended purpose(s). The intended purpose for Shaughnessy and Hazama is to correct the problem of Y2K ambiguity for date data stored in a database in a form that gives rise to the ambiguity, and Booth (as does Ohms) stores date data in a form that has no Y2K ambiguity. In the same way, the proposed combination would change the principle of operation of Shaughnessy and/or the combination of Shaughnessy and Hazama.

155. Contrary to the Examiner's assertion, and as indicated by the above discussion of Shaughnessy, Shaughnessy does not disclose "reformatting ... in order to facilitate collectively further processing the reformatted symbolic representations of each of the

symbolic representations of each of the dates." For this reason, the impropriety of the Examiner's rejection, as discussed above, is further supported.

156. Booth clearly and unequivocally notes that the dates stored in the database being manipulated by Clipper 5 are stored in integer format with a granularity of date days. As Booth notes at p. 939, "Dates are stored internally in such a way that math operations can be performed on dates to derive other dates. Adding an integer to date will result in a future date. Subtracting two dates will result in the number of days between the two." See also Booth at p. 99. Regardless of what Booth may say about date data entry, date display, or the like in the portions of Booth cited by the Examiner, the fact remains that the database of Booth does not use symbolic representations of dates in the Gregorian format, and does not have the Y2K ambiguity problem, since each date as stored is complete with information that indicates it YYYY characters in Gregorian format. Booth's disclosure of windowing in certain contexts, is, therefore, non-analogous art, teaches away from applicant's proposed solution to the Y2K date ambiguity problem, in the same way Ohms does. Further whatever sorting Booth proposes it is not of dates reformatted from a YY date data field representation into a YYYY format for purposes of being sorted in that format. Booth sorts in integer format simply by comparing the two integers with each other and the information contained in the integer itself is not Y2K ambiguous.

157. Ohms does not disclose:

> the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer implemented method for processing date[s] outside the twentieth century (see title, p 244 et seq).

The symbolic representations of dates in Ohms' database are in the form of a number representing a unique Lilian date and fully includes all year information such that the Y2K ambiguity problem addressed by the present invention is not even present vis-à-vis the date data stored in Ohm's database. Each Lilian date so stored includes (is convertible to) a representation of a full four character year, without Y2K ambiguity, just as is the case with the integer dates employed by Clipper 5 as described in Booth. In

addition Ohms does not call for the dates actually stored in the database to be within a ten decade window. In fact they can be anywhere within the span of days capable of being represented by seven digits (the specific embodiment disclosed, but it could be even more) of days, i.e., over 2000 years.

158.Ohms does not disclose:

> 'selecting a l0-decade window with a Y. sub. A Y. sub. B value for the first decade of the window, Y. sub. A Y. sub. B being no later than the earliest Y. sub. 1 Y. sub. 2 year designator in the database,' ... .

Ohms does disclose selecting a ten decade window for the windowing of date data being entered, but say nothing at all about selecting this ten decade window based upon any span of dates actually stored in the database.

159. Ohms is non-analogous art and actually teaches away from the present invention as claimed. Lilian dating does not suffer from the problems sought to be solved by the claimed invention, i.e., there is no Y2K date ambiguity that needs to be address in the processing of date data as stored by the Ohms system. While Ohms does disclose windowing and using a ten decade window, it is not in the context of the claimed invention. For example it is not for the purpose of "facilitating further processing of the dates [in the database]." Ohm's use of windowing is also not disclosed to be "reformatting the symbolic representation of the date [for each/all representations of dates stored in the database]." In addition Ohms doses not teach "sorting the symbolic representations of the dates" in a CCYYMMDD or like format, since dates in Ohms are sorted by comparison of the integer value that computes to the appropriate date, with the inclusion in the integer value of its four character year value. Indeed, modifying Hazawa, which teaches storing and manipulating (operating programs on) date data that is in, e.g., MMDDYY form, would render Hazawa inoperative for its intended purpose(s). The intended purpose for Hazawa is to correct the problem of Y2K ambiguity for date data stored in a database in a form that gives rise to the ambiguity and Ohms stores date data in a form that is not Y2K ambiguous. In the same way, the proposed combination of Ohms and Hazama would change the principle of operation of Hazawa.

160. There is no *prima facie* obviousness because there is no motivation to combine non-analogous art, especially where Ohms teaches away from the claimed invention, at least to the extent it teaches addressing the Y2K problem by storing date data in a form that is not susceptible to the Y2k ambiguity problem and does not need to be modified in any way to be able to be fully sorted, manipulated or otherwise processed without concern for any possible confusion between the Lilian value that represents all of the days throughout, e.g., the year 2002, including the fact that they are in the year 2002 and the totally unique and fully determinative integer value that represents each of the days in the year 1902, including the fact that each such day in within the year 1902, or, for that matter, 3002, 4002 and so on. The specific embodiment of Ohms using only seven characters to count the dates would have to be expanded to cover a span of more than 2000 years, however, and the starting date of the first day, e.g., January 1, 1900, would determine, along with the total number count of days, the end of the span that can be covered.

161. The Examiner is misreading the claims. As recited, e.g., in claim 33, the process step calls for:

> reformatting the symbolic representation of each symbolic representation of a date in the database, without changing any of the symbolic representations of a date in the database during the reformatting step, ...

or in claim 60 the process step calls for:

> by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; ... .

The "symbolic representations of dates in the at least one field of the database" remain unchanged, as discussed in further detail herein with regard to the enabling issue, as the

Examiner in the original application noted and as was the subject of an amendment of the claims therein to clarify that point.

162. As to written description, the Specification is reasonably clear in indicating that the applicant as of the filing date of the application leading to the Dickens patent was in possession of the invention, even if new terminology is used in the claims. The claimed sort based upon the reformatted CCYY format is a broader genus. As indicated in the specification and as would have been well known in the art at the time of the filing of the Dickens patent, data may be stored in databases in other than YYMMDD and in which the Y2K ambiguity problem still exists, e.g., YYMMMDD, where the MMM is a three letter designation of the month. For such date data formats, the present invention, as would be understood by those skilled in the art from the disclosure of the Dickens patent, with or without Exhibit A, is just as useful, along with the reformatting of the YY to CCYY. The claim, therefore, is a broader genus, which would cover the originally recited CCYYMMDD as well as, e.g., CCYYMMMDD.

163. The Examiner has taken the position that:

> Claims 33, 60-61, 64-65 and 70 call for reformatting to occur 'without changing' or 'without modifying' the symbolic date representations during the reformatting when the specification merely indicates that the YYMMDD date format is reformatted to appear in the form CCYYMMDD (col.3, lines 41-43). It is apparent that the original specification is devoid of any disclosure of how such reformatting is performed 'without changing' or 'without modifying' the symbolic date representation. In fact, the suggestion of reformatting without changing representation is on its face a contradiction, for the reformat is to change representation. Therefore, the claimed limitation reformatting to occur 'without changing' or 'without modifying' is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

164. There is quite an adequate disclosure in the original specification including the Certificate of Correction, from both a written description and an enabling perspective.

The addition of the Exhibit A further supports the claim language. The "without modifying" and/or "without changing" recitations refer to the fact that the original database date entry as contained in, e.g., a legacy database itself, is what is not modified. Clearly modification occurs according to the claims of the what is taken from or extracted from the date data field in, e.g., a legacy database, but this modification/reformatting according to the claims is done without also modifying/reformatting the originally stored date data as it is in the database itself and remains so after the converting and reformatting according to the claimed invention. This is adequately described in the written description and fully enabled.

165. The Examiner has taken the position that:

> Claims 16-30, 32, 34-67, 69-71, 75 and 76 call for processing relative to a 'pivot date' or 'pivot year' when such terms are nowhere defined or even mentioned in the original specification. Therefore, the claimed limitation 'pivot date' or 'pivot year' is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

166. This term is well known in the art. By way of example, United States Patent No. 6,317,746, entitled SOFTWARE DATE AND TIME SERVICES, issued to Franklin, Jr., et al. on November 13, 2001, and United States Patent No. 6,003,028, entitled IMPLEMENTING EXTENDED NUMERIC RANGE WITHIN A TWO-DIGIT SOFTWARE REPRESENTATION, issued to Koenig on December 14, 1999 use the term in connection with windowing techniques utilizing, e.g., a ten decade window. The Examiner has himself used the term throughout the prior and present Office Actions in rejecting claims with and without the term "pivot year" in the claim language. The term simply means, as the Examiner himself has used it, the starting year for the window.

167. The Examiner has also taken the position that:

> Claims 20-21, 62-65 and 71 call for 'reformatting' or 'storing' 'separately' from the symbolic representations in the database or from the database when the original specification merely suggests reformatting or sorting the date. However, the original specification does not disclose such 'separate' reformatting or storing.

Therefore, the claimed limitation of 'separate storing' or 'separate reformatting' is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

168. As discussed above with respect to claims 33, 60-61, 64-65 and 70 regarding the 'without changing' or 'without modifying' recitations, the Specification as originally filed along with the Certificate of Correction, adequately discloses and enables the recitations regarding "separate reformatting" and "separate storing." The addition of Exhibit A further supports such recitations.

169. The Examiner has taken the position that:

> Claims 16-25, 31-33, 66-67 and 72 call for 'collectively further processing' when the specification makes no mention of such 'collective' further processing. Similarly, claims 36-43 call for 'collectively sorting' or 'collectively manipulating' when the original specification merely suggests sorting and manipulating. However, it does not mention such 'collectively' sorting or manipulating. Similarly, claims 34-61, 63 and 65 call for the step of 'running a program collectively' when the original specification, perhaps, only implicitly discloses the 'running of the program'. However, such 'collective' running of the program, is not disclosed. Therefore, the claimed limitations of 'collective processing, 'collective sorting', 'collective manipulating' or 'collective running are new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

170. The original disclosure and claims disclose a process whereby "each" or "all" of the dates stored in a database, e.g., a legacy database, wherein the stored format includes only two year date characters, are reformatted to contain four year date characters, followed by a process of, e.g., sorting or manipulating, based on all of the reformatted dates. The Exhibit A disclosure further supports this interpretation of the claims. The term "collectively" is not used in the original disclosure. However the term serves to define over the art, e.g., Shaughnessy, where, e.g., one date from the database and one

fixed date, or two dates from the database, are compared to each other, in the called subroutine, as opposed to all of the data from the database being manipulated, e.g., date sorted "collectively."

171. The Examiner has taken the position that:

> F. Claims 36-37, 40-41, 48-49, 51-59, and 69 call for the running of a program after a sorting operation has been performed. However, the original specification does not provide a written description of such running of a program subsequent the step of sorting. Similarly, claims 38, 39, 42-43 call for data manipulation before running of the program. No written description is provided for such data manipulation before running the program in the original specification. Therefore, such limitations are new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

172. As noted in the Specification of the Dickens patent:

> Once the symbolic representations of the dates are reformatted according to the procedure set forth above, the date information may be sorted, numeral 38, or manipulated, numeral 40, together with the entries associated with the dates. Such manipulation may include handling of the data associated with the dates, storing the dates and associated information back in the data base, or other processes.

173. In addition, at least Exhibit A shows a sort program run before another program, e.g., a print program.

174. The Examiner has taken the position that:

> Claims 46-59 call for "repeating the step of converting at least a substantial portion" of the specified data. The original specification does not disclose the conversion of such substantial portion. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

175. A person skilled in the art at the time of the filing of the application leading to the Dickens patent would have understood from the disclosure of the Dickens patent, with or without Exhibit A, that the storage of databases, particularly of extensive nature, may be contained in memory in variously segmented ways, e.g., on pages of extended memory, or organized by, e.g., data entry number. In addition it would have been understood that the process of the present invention, depending upon the particular application program being utilized and the particular kind of "manipulation" being done, may effectively run on a substantially portion of the database containing a substantial portion of all of the, e.g., date data fields, but not necessarily all of them. Applicant's claims are not limited to only those instances where the recitation "each" or "all" as distinguishing over prior art, e.g., Shaughnessy, would require that each and every date data field is reformatted. In addition those skilled in the art would have appreciated that the database may contain several different date data fields associated with each particular data entry in the database and the sorting or other manipulation may only be concerned with one such field, and the conversion, therefore, only necessary in that instance and only as to that field. The program listing in Exhibit A is exemplary. For example, the "tools" database may have other date data fields besides "last_inv.dat", e.g., purchase.dat or last_maintenance.dat. The claims as originally filed would cover that situation and the added claims rejected above by the Examiner simply further clarify this fact.

176. The Examiner has taken the position that:

> Claims 34-65 and 70-71 call for 'converting' symbolic representations 'by windowing the symbolic representation' when the specification merely discloses the selection of a 10 decade window. The verb 'windowing' appears nowhere in the specification, and its meaning is unclear. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

177. The disclosure of the Dickens patent, even without Exhibit A, and also with Exhibit A, fully describes the claim recitation "by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against

a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database ... ." Windowing is a well known and recognized term in the art, and as noted above the pivot year, meaning the earliest date in the window, is also a well known and recognized term of art. Even if the disclosure of the Dickens patent, with or without Exhibit A, does not specifically use the term "windowing" or the term "pivot year," one skilled in the art at the time of the filing of the Dickens patent would have understood the disclosure to contemplate and fully describe and enable the claim limitation.

178. The Examiner has taken the position that:

> Claims 35, 37, 39, 41, 43, 45, 49, 51, 53, 55, 57 and 59 call for the step of 'opening the database prior to the step of converting' when the original specification makes no mention of opening the database. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

179. Applicant asserts that the step of opening the database is at least inherent in the disclosure of the Dickens patent. One skilled in the art would have understood that to get at the date data field stored in the database in the Y2K ambiguous format in order to reformat it to not be Y2K ambiguous, the database would initially have to be opened up for such access. Exhibit A, in addition, specifically includes a program step opening the "tools" database.

180. The Examiner has taken the position that:

> Claims *34-65, 70* and 71 call for the avoidance of an 'ambiguity' by reformatting or converting date representation. The original specification merely suggests that dates containing only two digit year representation, and without reformatting, may sort improperly. It does not mention or discuss any such claimed ambiguity. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

181. Applicant submits that there is a full description of and enablement of the claims recitation of a process for working on a "database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries," and for the subsequent recitation of "converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity ... ." The specification says that the problem being addressed is:

> However, with the turn of the century at Jan. 1, 2000, the representation and utilization of dates becomes more complex. Using the numerical form above, Dec. 15, 2000 is represented as 12/15/00. If a numerical sort is performed on 12/15/93 and 12/15/00, the later date 12/15/00 sorts as the first-occurring date, an incorrect result.
>
> Sets of dates spanning the turn of the century and associated with past, current, and future activities are now stored in many databases. When stored in the conventional formats discussed above, those dates will not readily be used and numerically sorted in chronological order.

In other words, because of the utilization of only two date data characters, the century of the date is ambiguous, and the process of the present invention will remove that ambiguity. That is, the date data format that is ambiguous in two characters is converted to one in four characters that is not ambiguous in a disclosed embodiment of the invention.

182. The Examiner has taken the position that:

> Claims *1-15, 31, 33, 68,* 72-74 call for the selection of a 'YAYB value for the first decade' of a window. There is no known meaning for the 'value of a decade' and the original specification is devoid of any description of what the 'value of a decade' is. Because this subject matter was in the original disclosure, such limitation is not new matter. However, it is rejected under the second paragraph of *35 USC 112* because the meaning of the claim phraseology is so devoid as to be wholly indefinite.

183. The specification and claims are perfectly clear on the meaning of the value of Y_AY_B. It is the "first year of the 10-decade window." (Col 3, line 13). The full recitation of the claim to which the Examiner refers recites "selecting a 10-decade window with a Y_AY_B value for the first decade of the window ... ." This is precisely the same as saying that the "value" of Y_AY_B is the two digit year value of the first year in the 10-decade window. Contrary to the Examiner's suggestion, the claim does not call for setting the "value for a decade," even if in the context of the Specification and claim language there would be any doubt that "the value of a decade" is ten years. The claim clearly calls for a "Y_AY_B value" which is "for the first decade of the [10 decade] window ... ." It is also the same value as for the first year in the 10-decade window.

184. The Examiner has taken the position that:

> 1. Applicant argues that Shaughnessy does not teach or suggest "*the step of selecting a 10-decade window YaYb no later than the earliest YI Y2 year designator in the database.*" Applicant alleges that Shaughnessy only discloses the selection of a 10 decade window utilizing the date the system was installed. In response, the Examiner respectfully submits that Shaughnessy teaches the selection of a 10-decade window in figure 4 and the necessity of such a window starting with a date no later than the earliest year in the database is taught in Hazama.

185. Shaughnessy does not disclose the claimed "step of selecting ... ." Shaughnessy in the Specification and in the discussion specifically of Figure 4, and in Figure 4 itself, does not disclose "YaYb no later than the earliest Y1Y2 year designator in the database." Specifically, depending upon the determination made in the block 36 of Figure 4 of Shaughnessy, the start date is set to either the "install date" or the "current date," and then the "100 year cycle" is determined by a selected number of years from the start date of so-called "future dating." This is explained by Shaughnessy as follows:

> FIG. 4 illustrates the steps performed to determine the end of the 100 year cycle. When a system is modified according to the principles of the present invention, several parameters may be specified. The parameters may include the number of

years of future dating required (default is 10), the type 2 format, CCYYMMDD, for the modified system install date (default is 19931231), and whether the end of the 100 year cycle is to be updated daily (0 indicates no update of the cycle, 1 indicates daily update of the cycle; default value is 1). The first step in determining the *end of the 100 year cycle* is therefore determining the update frequency for the cycle (box 36). If the cycle is to be updated daily, then the starting date is set to the current date (box 38), as determined above. Therefore, for this example the starting date would be 20000101 if the cycle is to be updated daily. Next, the *end of the 100 year cycle* is determined by adding the number of years of future dating required to the starting date (box 40).

Shaughnessy, at best, describes the selection of a desired end of the 100 year cycle, which indeed may be updated daily. Regard for the earliest date in the database is not considered. Both the install date and the current date may result in a 100 year window that will incorrectly translate dates in the database into the 21st century by Shaughnessy's disclosed method, or will at some point in time, if updated daily, begin to do so. Shaughnessy ignores or, at best, teaches away from the claimed process step of "selecting a 10-decade window YaYb no later than the earliest YI Y2 year designator in the database."

186. The Examiner has further taken the position that:

> 2. Applicant argues that neither Shaughnessy nor Hazama teaches or suggests *"the step of determining a century designator CI C2 for each symbolic representation of a date in the database, CI C2 having ...."* Applicant alleges that the teaching of Shaughnessy or Hazama is to determine a century designator for at most two date representations being processed in a called subroutine at a given time. In response to the preceding argument, the Examiner respectfully submits that even under the allegation above, the Shaughnessy-Hazama combination would still disclose the claimed limitation as long as the references teach or suggest the determination of a century designator for each date in the database. As discussed in the office action, Shaughnessy determines a century designator for converting a current date from a six-digit to an eight digit format before the converted date is returned for

use in a particular application. Shaughnessy determines the century value (19 or 20) by comparing the current date to the corresponding date portion when the system was installed with the modifications. Further, Shaughnessy suggests that the above approach can be used to determine a century designator for converting each six digit date in a database to corresponding eight digit dates. However, Shaughnessy refrains from such an approach, though capable of curing the year 2000 problem, on economic instead of technical grounds, since it might not be cost efficient. To the extent applicant is arguing that Shaughnessy fails to extrapolate the operation of date conversion from a single instance to an entire database, it is first noticed that one of ordinary skill in the art extrapolates single operations to batch processing of an entire database as a matter of automation efficiency, it is secondly pointed out that Shaughnessy teaches that its date conversion processing would be inserted for every occurrence of date processing, i.e. across the entire input gamut, col. 4 lines 27 to 33, and it thirdly noticed that Shaughnessy even provides a specific example of checking the due dates in a database for being overdue col. 4 lines 38 to 43. Further, Hazama complements Shaughnessy by disclosing the use of a pivot date that is smaller than any other date in the database to compare each date in the database with the pivot date to thereby determine whether each two digit year in the database should be preceded by 19 or 20. Therefore, the Shaughnessy -Hazama combination does teach the above limitation, as claimed.

187. The Examiner's combination of Shaughnessy-Hazama does not "teach or suggest the determination of a century designator for each date in the database." In the first place, neither reference expressly teaches performing the claimed process step on "each" date representation in the database. They teach calling up a subroutine if an application program encounters an ambiguous date representation. The fact that eventually the application program may encounter all of the dates, does not mean that the combination of the references teach performing in the specific sequence of process steps in the claims, first the reformatting of each of the dates and then sorting , manipulating, running a program, or the like, on them with respect to all of the date representations amounting to

"each of the date representation" as recited in the claims. In addition, even if the processes disclosed by Shaughnessy or Hazama or the combination of these references eventually could or might get to all date representations does not amount to a disclosure of the sequence of steps specifically recited in the claims regarding the "processing of symbolic representations of dates stored in a database," as to determining a century designator for each symbolic representation of a date in the database," followed by the step of "reformatting the symbolic representation of the [each such] date ... to facilitate further processing of the dates." While the claim does not recite "each such" this is implicit from the rest of the claim language and from the disclosure.[6]

188. The Examiner is incorrect to assert that:

> Shaughnessy determines a century designator for converting a current date from a six-digit to an eight-digit format before the converted date is returned for use in a particular application.

Shaughnessy returns a "parameter," which itself is not information from which the reformatting of the date data used to generate the "parameter" can be determined.

189. The Examiner is also not exactly correct in framing applicant's assertion with regard to what Shaughnessy suggests as to whether:

> the above approach can be used to determine a century designator for converting each six digit date in a database to corresponding eight digit dates. However, Shaughnessy refrains from such an approach, though capable of curing the year

[6] Furthermore, as noted above, those skilled in the art would have understood from the disclosure f the Dickens patent, with or without Appendix A, that "each" while it is distinguished from the processes of Shaughnessy and Hazama does not necessarily mean each and every possible date data stored in the data base. While that is most often the case in the operation of the claimed process, e.g., if the database is organized using, e.g., pages or sections of memory, and in the context of a given application program, "each" may mean each on a given page or in a given section and the application program may be able to deal with the reformatted group of "each of the date data representations," on a page by page or section by section basis, or may require reformatting of every one of the dates in every date data field having only two character date data information before continuing on the do the sorting of manipulating or otherwise run a program, after the process of the present invention has reformatted "each" of the date representations to facilitate further processing. Nevertheless, whether done on a page by page, sections by section, etc. basis or done throughout the entire database before "further processing," this method is distinguished from either Shaughnessy of Hazama or the combination of these references.

2000 problem, on economic instead of technical grounds, since it might not be cost efficient.

Shaughnessy's discussion comports with applicants, i.e., that to modify the existing legacy database is highly impractical, if not impossible. That is, as noted above, and in applicant's Specification, with or without Exhibit A, one does not want to change the legacy data base itself, e.g., its organization, data formats and sizes, etc. within, e.g., the date data fields, and/or with respects to, e.g., links, etc., employed in the data base, which might also have to be changed if a date data field is modified, e.g., enlarged to an entirely new date data field to accommodate, e.g., the expanded year date data containing the century designator. Shaughnessy proposes a solution, and Hazama proposes a solution, but they are not the solution of the claimed invention.

190. The Examiner's suggestion of "extrapolation," of Shaughnessy's approach into the claimed invention is hindsight reconstruction of the process disclosed in Shaughnessy. It is incorrect also for the Examiner to assert that this "extrapolation," amounts to simply multiplying the process steps proposed by Shaughnessy to cover the entire database:

> To the extent applicant is arguing that Shaughnessy fails to extrapolate the operation of date conversion from a single instance to an entire database, it is first noticed that one of ordinary skill in the art extrapolates single operations to batch processing of an entire database as a matter of automation efficiency ... .

To do so does not result in the claimed invention, since, at least, Shaughnessy does two by two comparisons (either of a date from the database and a fixed date or two dates from the database) and returns a "parameter" indicative of the results of that single two by two comparison. Even if multiplied over and over to go through the entire data base, it is still not the claimed process.

191. The Examiner is also incorrect to assert that:

> Shaughnessy teaches that its date conversion processing would be inserted for every occurrence of date processing, i.e. across the entire input gamut, col. 4 lines 27 to 33 ...,

or, even if correct, this is not the claimed invention, because Shaughnessy's "data conversion processing," as noted above, is not according to the claimed invention.

192. The specific portion of Shaughnessy referenced (and a continuing portion) discuss one embodiment of a process according to Shaughnessy's method in which:

> In accordance with the present invention, the current date operation routines nested in the body of the application program would be replaced with a call to one of a plurality of subroutines stored externally from the existing application program, as opposed to the date operation routine being reprogrammed to perform the date operation in a new format. The subroutines will be able to accommodate the date format currently employed by the application program, thus making it unnecessary to convert all of the date fields in files containing data used by the application program over to the new date format. For example, if an application program for a bank performed a date comparison to determine when loan payments were overdue, the point in the source code which previously performed the comparison may have program statements which performed the following functions:
>
> 1. Compare date next payment is due to today's date;
>
> 2. If the date next payment is due is greater than today's date, indicate that the account is OK.
>
> If the system which ran the above application program were modified in accordance with the principles of the present invention, then the program statements which performed the above functions would be modified to include program statements which did the following:
>
> 1. Call the subroutine which performs the date comparison passing today's date, the date next payment is due, and a three byte parameter, the first byte of which identifies the format of today's date, the second byte of which identifies the

format of the date next payment is due, and the third byte of which is left available for a return code indicative of a result of the comparison;

2. If the result received from the subroutine [indicated by the returned parameter] indicates that the date next payment is due is greater than today's date, indicate that the account is OK.

193. This is simply a very different process than the one recited in the claims, as noted above, even if performed over and over again to compare, e.g., sets of due dates to the current date as stored in the database, and as provided to the subroutine, until all of the entries in the database are examined. Therefore, while Shaughnessy may "provide[] a specific example of checking the due dates in a database for being overdue col. 4 lines 38 to 43," Shaughnessy does so by other than the claimed invention.

194. The Examiner has also taken the position that:

3. Applicant argues that neither Shaughnessy nor Hazama teaches or suggests the step of *'reformatting the symbolic representation of the date with the values C1C2, Y1Y2,MIM2, and DID2 to facilitate further processing of the dates.'* Applicant alleges that the teaching of Shaughnessy or Hazama is to reformat two dates at a time in the called [sic] result of the processing of the two reformatted date data entries, and not to facilitate further processing of the dates by reformatting the symbolic representations of the dates (claim 4). In response to the preceding argument, the examiner respectfully submits that the Shaughnessy-Hazama combination does disclose the reformatting of the dates in the C1C2Y1Y2M1M2D1D2 format to facilitate the further processing of these dates. Shaughnessy's conversion of the current date of an operating system from a six digit format to an eight digit format each time said date is going to be used in application. Such reformatted dates are further utilized by returning one date field with the converted date to the subroutine and by returning a parameter to the application program for use in further operations. As explained above, Shaughnessy suggests that such approach can be extended to reformat dates

already stored in database such that they can be used for further processing. Therefore, the ShaughnessyHazama combination does teach the above limitation, as claimed.

195. The above discussion of Shaughnessy is reiterated here, including, e.g., the discussion of the claimed invention dealing with "all" and "each" in the claimed process sequence resulting in the "facilitat[ion] of the further processing of the dates." In addition Shaughnessy does not:

> return[] one date field with the converted date to the subroutine and by returning a parameter to the application program for use in further operations.

and also does not:

> suggest[] that such approach can be extended to reformat dates already stored in database such that they can be used for further processing,

or, at least does not suggest doing so in the context of "facilitating the further processing of [each of] the dates ... ." Shaughnessy sends a date field to the subroutine and returns a parameter that is lacking in any indication of the date itself, whether as originally stored in the database or as converted by Shaughnessy within the subroutine for purposes of the functioning of the subroutine to create and return this parameter.

196. The Examiner has taken the position that:

> 5. Applicant argues that neither Shaughnessy or Hazama teaches or suggests the step of '*reformatting each symbolic representation of a date in a format CIC2YJ Y2M1M2DID2 (claim 5), nor sorting the symbolic representations of dates in numerical order sort (claim 6), nor storing the symbolic representation of dates and their associated information back into the database (claim 9); nor manipulating information in the database having reformatted date information therein (claim* 10).' In response to the preceding argument, the examiner respectfully submits that with regards to claim 5, Shaughnessy discloses the limitations as discussed above in paragraph 3 of the remarks. Regarding claim 6, Shaughnessy, Hazama and Booth disclose the cited limitation, see discussion above in paragraph 4 of remarks. Regarding the limitation of claim 9, Shaughnessy discloses the step of storing the symbolic representation of dates and their associated information back into the database, as discussed in the office

action. Shaughnessy teaches the storing in the database of current date after it has been converted from the six digit format to the eight digit format. Further, Shaughnessy suggests that such an approach can be extended to dates in a database. Consequently, Shaughnessy discloses the claimed limitation of claim 9. Regarding claim 10, Shaughnessy and Hazama disclose the cited limitations as discussed above in paragraphs 3 and 4 of the remarks.

197. In addition to the above discussion, Shaughnessy does not disclose:

the step of storing the symbolic representation of dates and their associated information back into the database, as discussed in the office action. Shaughnessy teaches the storing in the database of current date after it has been converted from the six digit format to the eight digit format.

Shaughnessy suggests that modifying the date data field in a legacy database and storing modified dates from an existing legacy database in such a modified date data field is possible but not practical. Such a modification is an undesirable alternative to the present invention as well, as noted above. The solution of the present invention is significantly different from the one suggested in Shaughnessy.

198. The Examiner has taken the position that:

14. Applicant argues that Booth does not teach or suggest the step of *'selecting YaYb such that Yb is 0 (zero),'* as recited in claim 8. Applicant alleges that even though SET EPOCH can and does use pivot years ending in 0, it is not a process according to the claimed invention. In response to the preceding arguments, the Examiner respectfully submits that Booth does disclose Yb to be zero by selecting YaYb to be equal to 90. See page 942. It is noted that Applicant's arguments that Booth's teaching is not a process according to the claimed inventions fails to comply with 37 CFR 1.111 (b) because they amount to a general allegation that the claim define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Applicant simply alleges that the cited limitations are not taught by Shaughnessy without actually explaining how these limitations are distinguishable from the corresponding portions in Booth on which the Examiner relied to establish the

prima facie case. See page ... of the office action. Consequently, Applicant has failed to successfully rebut the rejection of claim 8. Generally, Applicant bears the burden of explaining why the evidence on which the Examiner relies is insufficient to establish a prima facie case or demonstrating that Applicant has provided evidence which rebuts the prima facie case. See *In re Rouffet*, 149 F.3d 1350, 1355 47 USPQ2d 1453, 1455 (Fed. Cir. 1998). Furthermore, Shaughnessy's process would select a Yb value of 0 for one year out of every 10 when operated with daily update, col. 6. lines 4 to 45.

199. In addition to the above discussion, the fact that Booth may disclose the setting of a pivot year with a zero in the second place is not the claimed setting of the claimed Y_AY_B with a zero in the Y_B, because in the steps of the claimed process Y_AY_B is selected as earlier than the earliest date in the database, so that the claimed Y_AY_B is not just any pivot date but one selected as recited in the claim. As selected by Booth there is no disclosure of any regard being taken to the earliest date in the database. The same applies to the Examiner's reference to Shaughnessy, even if the fact Shaughnessy might sometimes unintentionally select a pivot year with a zero in the second place, depending upon the installation date or the updated installation date, or the like, is a disclosure of affirmatively carrying out the claimed step in the claimed process to so select the Y_AY_B, which it is not.

200. The Examiner has taken the position that:

> 15. Applicant argues that Ohms does not teach or suggest the step of *'Providing a database with symbolic representations of dates stored therein according to a format wherein M1M2 is the numerical month designator, D1D2 is the numerical day designator and YI Y2is the numerical year designator; all of the symbolic representations falling within a 10-decade period of time, as recited in claim* 1.' Applicant alleges that Ohms does not disclose the above limitations since Ohms teaches providing a database with the dates in a Lilian format. In response to the preceding arguments, the Examiner respectfully submits that Applicant's reading of Ohms is incorrect. Ohms teachings are not limited to dates in Lilian format. As

discussed in the office action, Ohms discloses the storing of dates in a database in Gregorian format, wherein said dates are converted from a six digit format (YYMMDD) to an eight digit format (YYYYMMDD). See page 247, table 1. Ohms further teaches that the dates stored in the database do fall within a ten decade period. See page 249. Consequently, the rejection is proper.

201. In addition to the above discussion, as noted Ohms does not disclose storing dates in the database in Gregorian format, and certainly does not disclose storing in Gregorian form any of the dates that are windowed for date data entry convenience. The entire point of Ohms is to avoid such a format in the storing of the date data by employing a Lilian format.

202. The Examiner has taken the position that:

> 17. Applicant argues that Ohms does not teach or suggest the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having ... .' Applicant contends that Ohm teaches entering date data into the database to be converted into Lilian format for storage and manipulation within the database. Applicant further alleges that since the conversion in Lilian format does not require the determination of a century designator for data in the database, then Ohms cannot teach such limitation. In response to the preceding arguments, the Examiner respectfully submits that Applicant misread Ohms' teachings. As pointed out above, Ohms' teachings are not limited to conversion in Lilian format. Ohms also discloses the conversion of dates stored in a database in Gregorian format from a six digit format to an eight digit format to include the century designator. See page 247, table I and page 248.

203. In addition to the above discussion, even if Ohms discloses "conversion," Ohms does not do so as to dates "stored in the database," since they are in need of no such conversion, and does not do so for purposes of "facilitating further processing of the dates."

204. The Examiner has taken the position that:

> 18. Applicant argues that Ohms does not teach or suggest the step of 'reformatting the symbolic representation of the date with the values C 1 C2, Y 1 Y2, M 1 M2, and D 1 D2 to facilitate the further processing of the dates.' Applicant contends that Ohms does not disclose such limitation since it teaches reformatting into Lilian format and thereafter processing the date data in the database utilizing the Lilian format. In response to the preceding arguments, the Examiner respectfully submits that, as pointed out above in the remarks, Ohms' teachings are not limited to reformatting in Lilian format. Ohms discloses the reformatting of a short Gregorian date having six digit into a Gregorian date having eight digits. See page 247, table 1.

205. In addition to the above discussion, even if Ohms performs "reformatting" it is not of the "dates stored in the database" and it is not for the purpose of "further facilitating processing of the dates," since Ohms facilitates processing of the dates by storing them in the database in Lilian format.

206. The Examiner has taken the position that:

> 19. Applicant argues that Ohms does not teach or suggest the steps of sorting the symbolic representations of dates (claim 4); or reformatting each symbolic representation of a date into the format C1C2Y1Y2M1M2D1D2 (claim 5) or sorting the symbolic representations of dates and their associated (claim 6) or storing the symbolic representation of dates and their associated information back into the database (claim 9) or after the step of reformatting, manipulating information in the database having the reformatted date information therein (claim 10) or converting pre-existing date information having a different format into the format wherein M1M2 is the numerical month designator, D1D2 is the numerical day designator and YlY2 is the numerical year designator or selecting YaYb such that Yb is 0 (zero) (claim 8). In response to the preceding arguments, the examiner respectfully submits that it was conceded in the office action that Ohms does not teach the limitations of claims 4, 6, and 8. However, Booth was relied

upon to complement Ohms' teachings in order to reject the cited claims. Regarding claims 5, 9 and 10, it was pointed out in the office action that Ohms teaches the reformatting of short order Gregorian dates having six digits into Gregorian dates having eight digits to thereby store the converted dates in the database for further use and processing. The limitations of these claims were fully addressed in the office action. It is noted, however, that Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Applicant simply alleges that the cited limitations are not taught by Ohms without actually explaining how these limitations are distinguishable from the corresponding portions in Ohms on which the Examiner relied to establish the prima facie case. See page ... of the office action. Consequently, Applicant has failed to successfully rebut the rejection of claims 4-10 as laid out in paragraph ... Generally, Applicant bears the burden of explaining why the evidence on which the Examiner relies is insufficient to establish a prima facie case or demonstrating that Applicant has provided evidence which rebuts the prima facie case. See *In re Rouffet*, 149 F.3d 1350, 1355 47 USPQ2d 1453, 1455 (Fed. Cir. 1998).

207. In addition to the above discussion, Ohms' does not reformat date data having six characters into "Gregorian dates having eight digits *to thereby store the converted dates in the database for further use and processing* ... ." Ohms stores dates in the database in Lilian format for further use and data processing. At least this element of the claims is missing from the combination relied upon by the examiner to find *prima facie* obviousness. In addition, as noted above, Ohms does not reformat dates "stored in the database," from the recited YY format to the recited CCYY format, and does not do any such reformatting for purposes of "facilitating further processing of the dates."

208. One skilled in the art, given the disclosure that the invention would facilitate a sort of dates in a format of $C_1C_2Y_1Y_2M_1M_2D_1D_2$, would realize that sorting of just $C_1C_2Y_1Y_2$,

without regard to also sorting the month and day data, is possible, without undue experimentation.

209. The specification and claims of the Dickens patent support the recitations of claims 36-37, 40-41, 51-59, 69 and 38, 39, and 42-43. This is further evidenced by the content of Exhibit A. Exhibit A includes instructions that amount to "collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations" (claims 36, 37) and "collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations" (claims 40-41), and "collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations" (claims 48-49) and "collectively manipulating the converted symbolic representations" (claims 50-51) and "collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program" (claims 52-53) and "collectively manipulating the converted symbolic representations" (claims 54-55) "wherein the program performs an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date" (claims 56-59) and "reformatting the symbolic representation of each symbolic representation of a date in the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C_1 C_2, Y_1 Y_2; sorting the

reformatted symbolic representations of the dates in the form C_1 C_2 Y_1 Y_2; and running a program on the reformatted symbolic representations of each of the dates" (claim 69) and "collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations" (claims 38-39) and "collectively manipulating the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations" (claims 42-43). Exhibit a shows that the data being sorted includes the reformatted dates.

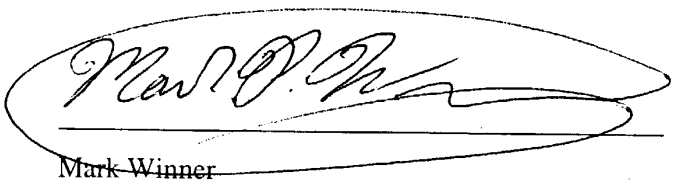
210. Exhibit A supports the claimed process steps of "collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates" (claims 16-25) and "collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates" (claims 31-33) and "to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates" claims 66-67) "prior to collectively further processing information contained within the database associated with the respective dates" (claim 72) and "collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations" (claims 36-37) and "collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations" (claims 38-39) and "collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations" (claims 40-41) and "collectively manipulating the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations" (claims 42-43).

211. Exhibit A supports the recitations of "reformatting each symbolic representation of a date into the format C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2 separately from the symbolic representations in the database" (claims 20-21) and "running a program on the stored converted symbolic representations to sort or otherwise manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database" (claims 62-65) and "running a program on the stored converted symbolic representations of each of the converted symbolic representations of the dates to sort or otherwise manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database" (claim 71).

212. The Dickens patent Specification, even without Exhibit A, but also including Exhibit A, supports the recitation "converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting" (claims 34-59) and "converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting" (claims 60-65). Specifically the term "windowing" as used in these claims is used according to the ordinary meaning that one of ordinary skill in the art would understand in the context of the other claim recitations of these claims.

213. I Mark Winner, make the above statements based upon my own knowledge and experience as one skilled in the art at the time of the filing of the application leading to the Dickens patent and based upon my review and understanding of the documents referenced in Paragraph 2 above. I make these statements of my own knowledge, or if based upon information and belief, based upon my so being informed and believing the statement to be true. I make these statements with the knowledge that willful false statements are punishable by fine or imprisonment under 18 U.S.C. §1001 and like statutes and laws and that such willful false statements may jeopardize the validity of the above referenced application and any patent(s) that may arise from this proceeding.

Respectfully submitted,



Mark Winner

[redacted] [redacted]

Mark Winner - Senior Systems Software Engineer - Computer consulting for the past 25Years

Boeing / McDonnell Douglas Corp / MDTSC
1/95 to present

In my current contract with Boeing the following business accounting systems have been my responsibility; Cost Charge Number System, Work Authority System, Merit Review System, Integrated Graphics Load, Labor Accounting and Reporting, Contract Status Information System, Manual Journals Vouchers, Rate Management System, Affordable Staffing, and many more.

A large part of my responsibility is for the migration of applications from old architecture to new Common Boeing Systems and new Web based applications.

Part of this migration effort was to convert the HR systems to People Soft. I was involved in the selection and evaluation of the People Soft package, and the conversion and migration of applications to the Web. Labor accounting, cost charge number maintenance system, and reporting were redeveloped and deployed onto the Web. The department matrix for the company reorganization during the Rockwell/Boeing merge was also converted.

Year 2000 Project oversight for accounting applications;

This included the initial impact estimates of the project scope for accounting applications of the Expendable Launch Systems Division. Manpower estimates and requirements for project scope within corporate guidelines were provided. Also various organizations were worked with to determine if application processes could be achieved by other manual methods or migrated to other organizational Year 2000 compliant systems. Business critical and non-business critical systems with year 2000 risk had remediation plans written. Project statuses were managed, reported, and presented.

Hewlett Packard system administration and maintenance of accounting applications for the Expandable Launch Systems;

Systems and infrastructure were maintained, and users were supported with their ongoing needs. Support efforts were made to migrate and replace applications and hardware with newer technologies by migrating to Web based applications and client/server based applications were more cost effective. Technical support was provided by working with users in the system selection process or design and development of new Web based applications.

The above activities utilize the following languages: Cold Fusion, Java, Java Script, HTML, Dreamweaver, Visual Basic, Microsoft Basic, Access, Delphi, Crystal Reports, XML, Versata, ASP, Perl, ftp/tcpip/http, Versata, and Rational Rose UML.

Hardware and operating system platforms: NT, HP-UX/UNIX, IBM-AIX, HP3000, IBM 3090's.
Other Languages: Basic, Cobol, Fortran, ALC, TSO, VMS, CICS, Cognos, Protos, C, PL1, Pascal, SPF.
Data Bases: Oracle 7.3, 8.0,8I; Microsoft SQL Server; DB2, Access DB.

Application Packages: People Soft.
Development tools: Rational Rose, Versata.
Testing Tools: Win-Runner, Rational Robot.

[redacted]

Western Digital
1/95 – 1/96

Responsibilities included the creation of a new process where by financial reporting was automated to be more accurate, timely, and better able to meet the users' needs. This caused an old batch reporting system to become an online documents' retrieval system enabling users to review organizational reports. This allowed each organization to customize their own month end reporting requirements. The result eliminated the need for hard copies because the users could get online information.

Beach Street Medical Insurance.
3/93 – 1/95

Responsibilities included system support and network administration. Duties encompassed support of all applications including Physician Billing Systems, Provider System, and Marketing System. Customers were closely worked with to integrate their data into the central repository. The external clients had a wide variety of PC hardware and software supporting their data that was loaded into the data repository on an ongoing basis. Because customers were responsible for providing the data, insurance of data integrity needed to be confirmed. Then repricing the billing and sending back the data in completed customer format was the final task.

Conversion of above mentioned systems were to the IBM AS400 Platform.
Tools used where COBOL, Image/Query, Reflections, Quick, Quiz, QTP, Supertool, and Novell Networking. Systems were on NetWare 3.12 and later upgraded to 4.0x

McDonnell Douglas Corp.
2/92 – 3/93

Responsibilities were maintenance of the Human Resources System including Compensation, Employee Biography, Employee Verification, Absence Tracking, Loan Verification, Employee Rate History, Ride Share, Merit Review, Seniority Training, Classified Materials Management System, and Medical Examination System.

These systems were developed in Cobol and supported and enhanced with Supertool, Qedit, MPEX, Nbspool, Adager, MRJE, Allbase/SQL, Image/Query, NS3000 Communications, and FTP. All above systems were network interfaced to be accessible to both local LAN systems and company corporate WAN's.

Tools used included Novell NetWare 3.12, 4.00, 4.01. All systems were multi-server and fault tolerant. Communications were via dial-up and lease lines including TCP/IP.

Coldwell Banker
2/91 – 1/92

Responsibilities were to maintain their Affiliate Franchise System. This System was written to run on a HP3000 using Cobol, Image Database, and Quiz. The system was converted to run on an IBM 3090 under VMS using DB2/SQL, and CICS. The work environment was on PS2's use through a Novell Network. This was done by using Reflections to link into the HP, and PC3270 to link into the IBM mainframes on the other coast.

Pacific Mutual
9/90 – 2/91

Responsibilities were to design a system to replace an installed Financial Administration System, and Annuity Payment System. The new system was to provide a more cost effective processing of mutual funds. The vender selection process and the installation and implementation of the new systems were achieved.

Western Digital
3/88 – 9/88

Responsibilities were to take over the development of the booking and backlog systems in order to provide a more accurate and timely reporting of international sales from (Japan, Germany, United Kingdom, and France). Remote systems were accessed using Telnet, through a local area network and WAN.

Pacific Lighting Energy Systems
12/87 – 03/88

Responsibilities were to convert their existing accounting package from an HP3000 to a PS2 Novell Network. Through the conversion the cost tracking method was changed and this restructured the chart of accounts to allow for many more cost centers. Five other supporting subsystems: accounts payable, labor collection, profit and loss, work order, and operational maintenance were included.

Interstate Electronics
3/87 – 10/87

Responsibilities were to upgrade their Boeing MFG package, then to implement the purchasing /receiving modules for the system, while maintaining the system on a continuous basis, with a close user interface to meet their specific needs.

Hughes Aircraft
7/86 – 3/87

Responsibilities were to implement enhancements and conduct maintenance on four systems. Labor Collection, Journal Voucher System, Automatic Cost Transfer System and Automated System Controls. These systems were functioning in a distributed fashion on eight HP3000 systems with centralized processing on an IBM 3083. Other responsibilities were to describe their Contract life cycle from award through final contract closure. This was done using the SDM/70 methodology, and the ADG.

Litton Systems
2/86 – 7/86

Responsibilities were to take over the requisitioning function of the Supply and Maintenance System for the Saudi Air Force. Completed and maintained the Design Specifications, reviewed the Protos and Cobol code, managed unit and integration testing to insure that the system meet the requirements, and reflected the ongoing changes.

Business Logic
5/85 – 2/86

Product market analysis was made of a potential over sea's distribution package for municipal utilities to handle accounting, financial analysis and investment tracking, and billing needs. This system utilized the HP3000 system using the following software features: Cobol, Basic, SPL, MRJE and MPE.

Hughes Division 15
7/84 – 7/85

Managed the conversion of the Quality Information System from compatibility with the Amdahl/370 Production Management System. Designed and implemented a system for converting Software and JCL from MPE-IV to MPE V operation system. Developed schedules and managed the conversion.

Hughes Division 64
8/83 – 7/84

Responsibilities were to design and implement a Contract Awards System. Also the design of a Contract Information Retrieval System utilizing Hughes Application Development Guide (ADG), other duties included major enhancements to the Maintenance and Project Control System (MPC).

Northrop
12/82 – 7/83

Responsibilities at this major aerospace corporation included extensive systems analysis to determine requirements and specifications for the new system as well as making recommendations for enhancements to existing systems. The position required heavy user interface on a daily basis. Also managed system testing and implementation with the users. Applications were all on the HP3000 environment using Cobol.

Farr Company
8/81 – 11/82

Responsibilities included the design and implementation of a new online parts pricing system and an MRP system. As well as the implementation of a sales order entry system. Provided management of the programmers performing maintenance. Interfaced with users to develop specs for programmers. Implemented a BOM system that interfaced with the MRP system.

R. J. Hanson
8/80 – 7/81

Responsibilities were to direct development and implementation of a Supply Information Purchasing System for the Province of Alberta, Canada. The System served as the central source of purchasing and associated purchasing strategies for the Province. The system included source of supply, direct purchase orders, requisitions, standing offer support, and scheduled purchases. The online system specs were developed to run on a VAX 7700, or an HP3000 once the hardware platform was selected.

Information Handling Service
1/80 – 6/80

A software and hardware system was designed and implemented to handle document correspondence and tracking between divisions of the Department of Water and Power. Various hardware and software platforms were involved.

Applications Data Enterprises
5/77 – 1/80

The design, development and installation of systems and hardware was maintained.. Applications included MFG, Distribution, Shop Order Scheduling, MRP, BOM, Production Control, Inventory, General Ledger, Accounting, Payroll, and Personnel. As DP Manager all aspects of operation were overviewed. There were two programmers, and two operations staff. Direction was taken from the owners and customers.

Basic Four Corporation
2/75 – 5/77

Designed and programmed Demo systems to support the sales staff. Developed educational classes and training programs. Also aided Sorbus on hardware problems and handled problem accounts. Basic Four hardware and software version 1 through 4 were used.

Burke – Winner Inc.
1/77 –6/80

As a partner in this consulting firm, responsibilities included dealing with upper and lower management in solving hardware and software needs and problems. Managed the design, implementation, debugging, and customizing of applications to meet the client's requirements. Applications included Distribution, Order Processing, Business and Manufacturing applications.

Newport Mesa Unified School District
1/74 – 2/75

Responsibility was taken for the application programming of a new personnel payroll system and student records system. The application program was developed and completed documentation given.